

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018



A Mini-Project Report On

INTERNAL RESULT MANAGEMENT SYSTEM

*Submitted in partial fulfillment of the requirements as a part of the DBMS Lab for the award
of degree of*

Bachelor of Engineering in Information Science and Engineering

Submitted by

CHETAN N S
1RN16IS027

Faculty Incharge

Mr. Manoranjan S R

Asst. Professor

Dept. of ISE, RNSIT

Lab Incharge

Mr. R Rajkumar

Asst. Professor

Dept. of ISE, RNSIT



Department of Information Science and Engineering
RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,

Bengaluru – 560 098

2018 – 2019

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the mini project report entitled **INTERNAL RESULT MANAGEMENT SYSTEM** has been successfully completed by **CHETAN N S** bearing USN **1RN16IS027** presently V semester student of **RNS Institute of Technology** in partial fulfillment of the requirements as a part of the DBMS Laboratory for the award of the degree *Bachelor of Engineering in Information Science and Engineering* under **Visvesvaraya Technological University, Belagavi** during academic year 2018 – 2019. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements as a part of DBMS Laboratory for the said degree.

Mr. Manoranjan S R
Faculty Incharge

Mr. R Rajkumar
Lab Incharge

Dr. M V Sudhamani
Professor and HOD

External Viva

Name of the Examiners

Signature with date

1. _____

2. _____

ABSTRACT

The main purpose of Internal Result Management System is used to automate the existing manual system to full pledged computer software, fulfilling their requirements so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Internal Result Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on other activities rather than on record keeping. Thus it will help organisation in better utilization of resources. The organisation can maintain computerized records without any redundant entries.

Efficient implementation of Internal Result Management System can save a lot of time when compared to manual record keeping of the results. It also makes calculations of the average marks easier and error free since calculations are not manual instead are carried out by a software itself.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

I would like to profoundly thank **Management of RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

I would like to thank our beloved Director **Dr. H N Shivashankar** for his confidence feeling words and support for providing facilities throughout the course.

I would like to express my thanks to our Principal **Dr. M K Venkatesha** for his support and inspired me towards the attainment of knowledge.

I wish to place on record my words of gratitude to **Dr. M V Sudhamani**, Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind my Project work.

I would like to express my profound and cordial gratitude to my Faculty incharge **Mr. R Rajkumar**, Assistant Professor, Department of Information Science and Engineering for his valuable guidance, constructive comments and continuous encouragement throughout the Project work.

I would like to express my profound and cordial gratitude to my guide **Mr. Manoranjan S R**, Assistant Professor, Department of Information Science and Engineering for his valuable guidance in preparing Project report.

I would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, I would hereby acknowledge and thank my parents who have been a source of inspiration and also instrumental in carrying out this Project work.

CHETAN N S

1RN16IS027

TABLE OF CONTENTS

CERTIFICATE	
ABSTRACT	i
ACKNOWLEDGMENT	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	iv
1. INTRODUCTION	1
2. E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM	2
3. SYSTEM DESIGN	4
3.1 Tables Description	4
4. IMPLEMENTATION	7
4.1 Requirements	7
4.2 Front End and Back End	8
4.3 Code Discussion	10
4.4 Snapshots	15
5. CONCLUSION AND FUTURE ENHANCEMENTS	20
REFERENCES	21

LIST OF FIGURES

Fig.No	Descriptions	Page
Fig 2.1	Entity-Relationship Diagram	2
Fig 2.1	Relational Schema Diagram	3
Fig 3.1	Login Table	4
Fig 3.2	Teacher Table	4
Fig 3.3	Class Table	4
Fig 3.4	Student Table	5
Fig 3.5	Parents Table	5
Fig 3.6	Results Table	5
Fig 3.7	Avgmark Table	6
Fig 3.8	Subjects Table	6
Fig 4.1	Login page	15
Fig 4.2	Change Password page	15
Fig 4.3	Teacher page	16
Fig 4.4	Student details entry page	16
Fig 4.5	Students details are Successfully	17
Fig 4.6	Select Semester of the Results to Insert	17
Fig 4.7	Insert the Results	17
Fig 4.8	Results are Successfully Inserted	18
Fig 4.9	To connect Parents and Students	18
Fig 4.10	Page when the a Parent Login	19
Fig 4.11	Page when a Student Login	19

Chapter 1

INTRODUCTION

“INTERNAL RESULT MANAGEMENT SYSTEM”, is a software application designed in order to keep a record of test performances by the students of an Educational institution. The project aims in reducing the difficulties faced in the existing systems and to provide a smooth, error free

Working mechanism which calculates the average of the test marks and makes it to be viewed by students and their parents.

There are three different users for this software:

A teacher can enter the email and password and login to the Examination result management system software. On entering the system software, a teacher can add student details, link USN of a particular student with their parents email id and can update the marks secured by a student in each test

Parent can enter the email and password and login to the Examination result management system software. On entering, a list of his/her ward is shown studying in the institution. Parent can click and view the result of his/her children.

A student can enter the email and password and login to the Examination result management system software. On entering he/she can view their marks secured in the tests conducted by the institution.

Final average marks can also be viewed by the students after the teacher has finished updating all the three test marks.

Chapter 2

E R DIAGRAM AND RELATIONAL SCHEMA DIAGRAM

2.1 Entity-Relationship Diagram

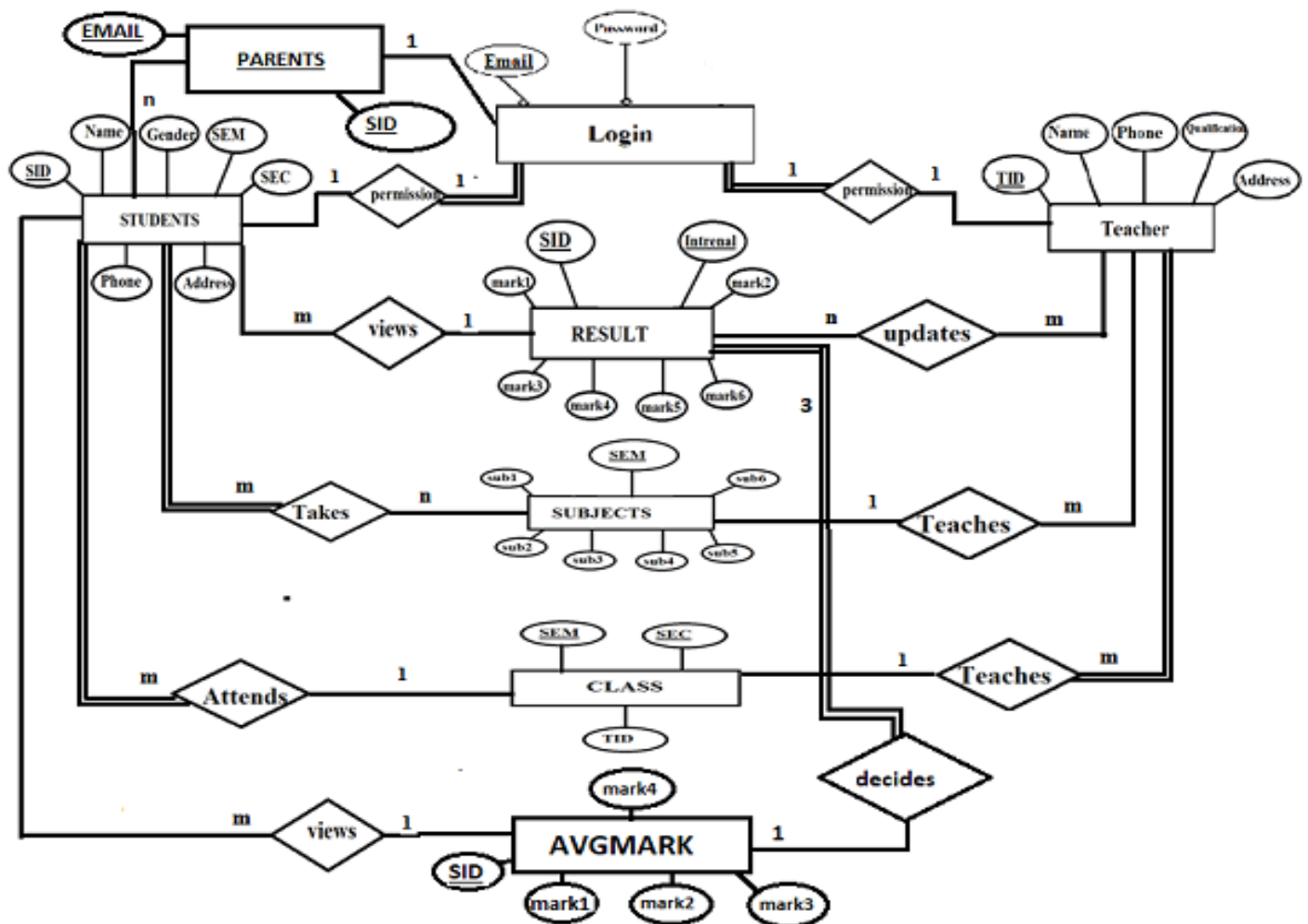


Fig 2.1(ER Diagram)

Entity-relationship modeling was developed for database design by Peter Chen and published in a 1976 paper. However, variants of the idea existed previously. Some ER models show super and subtype entities connected by generalization-specialization relationships and an ER model can be used also in the specification of domain-specific ontologies

An entity-relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form, for example: one building may be divided into zero or more apartments, but one apartment can only be located in one building.

2.2 RELATIONAL SCHEMA DIAGRAM

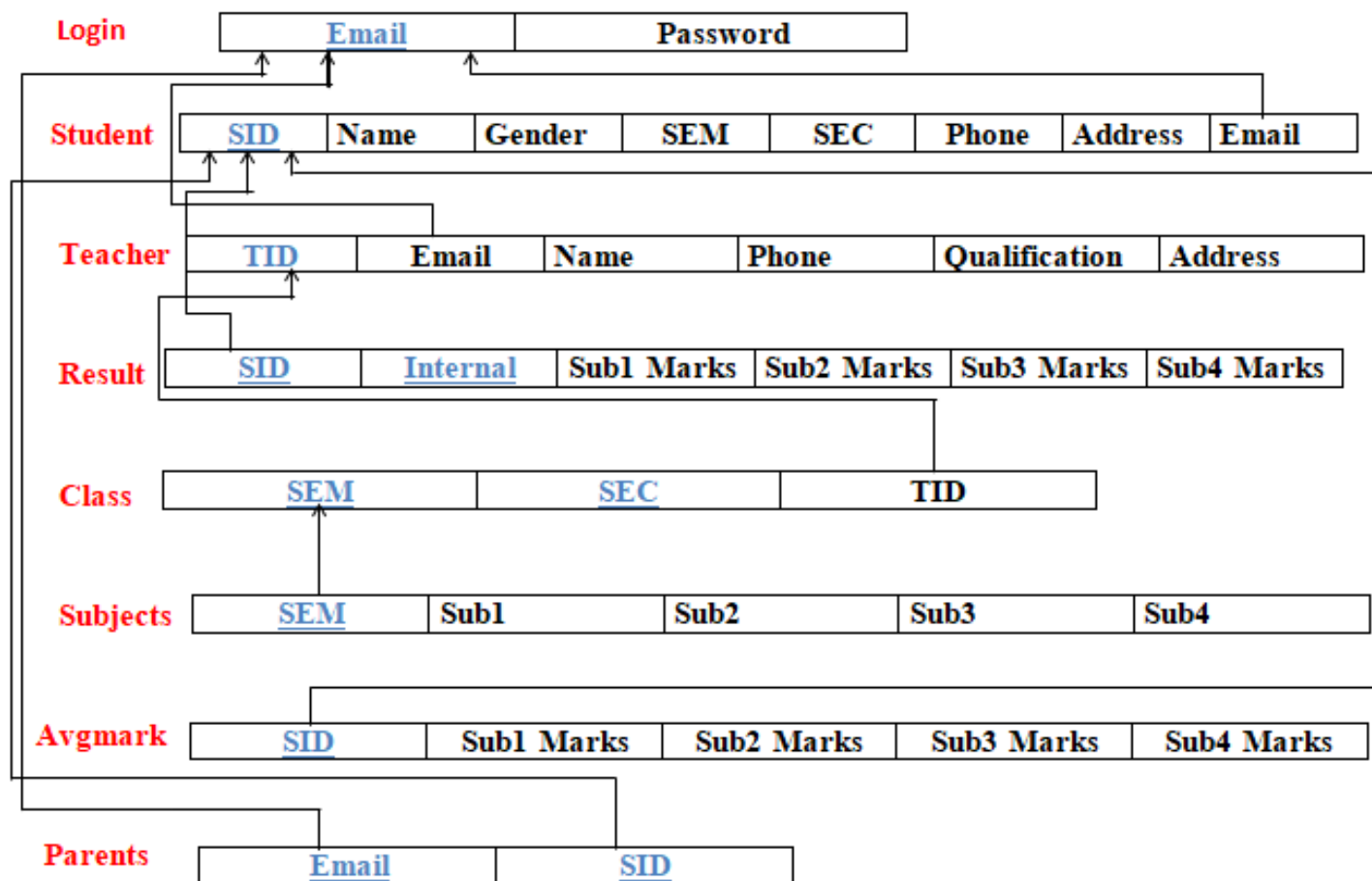


Fig 2.2 Relational Schema diagram

A Schema is a pictorial representation of the relationship between the database tables in the database that is created. The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modeled in the database.

Chapter 3

SYSTEM DESIGN

3.1 Tables

- **Login table**

```
CREATE TABLE LOGIN(  
EMAIL VARCHAR(50) PRIMARY KEY,  
PASSWORD VARCHAR(20),  
);
```

```
SQL> DESC LOGIN;  
Name                               Null?   Type  
-----  
EMAIL                             NOT NULL VARCHAR2(50)  
PASSWORD                          VARCHAR2(20)  
SQL> _
```

Fig 3.1 Login table

- **Teacher table**

```
CREATE TABLE TEACHER(  
TID VARCHAR(10) PRIMARY KEY,  
NAME VARCHAR(20),  
PHONE NUMBER(13),  
QUALIFICATION VARCHAR(20),  
ADDRESS VARCHAR(50),  
EMAIL VARCHAR(50) REFERENCES LOGIN(EMAIL)  
);
```

```
SQL> DESC TEACHER;  
Name                               Null?   Type  
-----  
TID                             NOT NULL VARCHAR2(10)  
NAME                           VARCHAR2(20)  
PHONE                          NUMBER(13)  
QUALIFICATION                  VARCHAR2(20)  
ADDRESS                        VARCHAR2(50)  
EMAIL                          VARCHAR2(50)  
SQL>
```

Fig 3.2 Teacher table

- **Class Table**

```
CREATE TABLE CLASS(  
SEM NUMBER(1),  
SEC CHAR, PRIMARY KEY(SEM,SEC),  
TID VARCHAR(10) REFERENCES TEACHER(TID) ON DELETE SET NULL  
);
```

```
SQL> DESC CLASS;  
Name                               Null?   Type  
-----  
SEM                             NOT NULL NUMBER(1)  
SEC                             NOT NULL CHAR(1)  
TID                             VARCHAR2(10)  
SQL> _
```

Fig 3.3 Class table

- **Students Table**

```
CREATE TABLE STUDENTS(
  SID VARCHAR(10) ,
  NAME VARCHAR(20),
  GENDER CHAR,
  SEM NUMBER(1),
  SEC CHAR,
  PHONE NUMBER(13),
  ADDRESS VARCHAR(50),
  EMAIL VARCHAR(25) REFERENCES LOGIN(EMAIL) ,
  PRIMARY KEY(SID)
);
```

```
SQL> DESC STUDENTS;
Name                               Null?    Type
-----
SID                                NOT NULL VARCHAR2(10)
NAME                               NOT NULL VARCHAR2(20)
GENDER                             CHAR(1)
SEM                                NUMBER(1)
SEC                                CHAR(1)
PHONE                              NUMBER(13)
ADDRESS                            VARCHAR2(50)
EMAIL                              VARCHAR2(25)
```

Fig 3.4 Student table

- **Parents Table**

```
CREATE TABLE PARENTS(
  EMAIL VARCHAR(50) REFERENCES LOGIN(EMAIL) ON DELETE
  CASCADE,SID VARCHAR(10) REFERENCES STUDENTS(SID),
  PRIMARY KEY(EMAIL,SID)
);
```

```
SQL> DESC PARENTS;
Name                               Null?    Type
-----
EMAIL                              NOT NULL VARCHAR2(50)
SID                                NOT NULL VARCHAR2(10)
```

Fig 3.5 Parents table

- **Results Table**

```
CREATE TABLE RESULTS(
  SID VARCHAR(10) REFERENCES STUDENTS(SID) ON DELETE
  CASCADE, INTERNAL NUMBER(1), MARK1 NUMBER(4),
  MARK2 NUMBER(4), MARK3 NUMBER(4),
  MARK4 NUMBER(4), MARK5 NUMBER(4),
  MARK6 NUMBER(4), PRIMARY KEY(SID,INTERNAL));
```

```
SQL> DESC RESULTS;
Name                               Null?    Type
-----
SID                                NOT NULL VARCHAR2(10)
INTERNAL                           NOT NULL NUMBER(1)
MARK1                              NUMBER(4)
MARK2                              NUMBER(4)
MARK3                              NUMBER(4)
MARK4                              NUMBER(4)
MARK5                              NUMBER(4)
MARK6                              NUMBER(4)
```

Fig 3.6 Results table

- **Avgmark Table**

```
CREATE TABLE AVGMARK(
  SID VARCHAR(10) REFERENCES STUDENTS(SID) ON DELETE
  CASCADE,
  MARK1 NUMBER(4),
  MARK2 NUMBER(4),
  MARK3 NUMBER(4),
  MARK4 NUMBER(4),
  MARK5 NUMBER(4),
  MARK6 NUMBER(4),
  PRIMARY KEY(SID)
);
```

```
SQL> DESC AVGMARK;
Name                               Null?    Type
-----
SID                                NOT NULL VARCHAR2(10)
MARK1                             NUMBER(4)
MARK2                             NUMBER(4)
MARK3                             NUMBER(4)
MARK4                             NUMBER(4)
MARK5                             NUMBER(4)
MARK6                             NUMBER(4)
```

Fig 3.7 Avgmarks table

- **Subjects Table**

```
CREATE TABLE SUBJECTS(
  SEM NUMBER(1) ,
  SUB1 VARCHAR(20),
  SUB2 VARCHAR(20),
  SUB3 VARCHAR(20),
  SUB4 VARCHAR(20),
  SUB5 VARCHAR(20),
  SUB6 VARCHAR(20),
  PRIMARY KEY(SEM)
);
```

```
SQL> DESC SUBJECTS;
Name                               Null?    Type
-----
SEM                                NOT NULL NUMBER(1)
SUB1                              VARCHAR2(20)
SUB2                              VARCHAR2(20)
SUB3                              VARCHAR2(20)
SUB4                              VARCHAR2(20)
SUB5                              VARCHAR2(20)
SUB6                              VARCHAR2(20)
```

Fig 3.8 Subject table

Chapter 4

IMPLEMENTATION

4.1 Requirements

The requirements can be broken down into 2 major categories namely hardware and software requirements. The former specifies the minimal hardware facilities expected in a system in which the project has to be run. The latter specifies the essential software needed to build and run the project

4.1.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - Intel i3 or better
- Processor Speed - 500 MHz or above
- RAM – 1GB or above
- Storage Space - Approx. 1GB

4.1.2 Software Requirements

- Technology Implemented : Apache Tomcat Server
- Language Used : JAVA(J2EE)
- Database : Oracle
- User Interface Design: HTML ,Servlets
- Web Browser: Google Chrome
- IDE : Eclipse

4.2 Front End and Back End

4.2.1 HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `` and `<input>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

4.2.2 CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

4.2.3 SERVLET

A Java servlet processes or stores a Java class in Java EE that conforms to the Java Servlet API,^[1] a standard for implementing Java classes that respond to requests. Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP. Thus "servlet" is often used as shorthand for "HTTP servlet". Thus, a software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML and more commonly, JSON. Servlets can maintain state in session variables across many server transactions by using HTTP cookies, or URL mapping.

To deploy and run a servlet, a web container must be used. A web container (also known as a servlet container) is essentially the component of a web server that interacts with the servlets. The web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

4.2.4 ECLIPSE

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

4.2.4 Apache Tomcat Server

Apache Tomcat, often referred to as Tomcat Server, is an open-source Java Servlet Container developed by the Apache Software Foundation(ASF). Tomcat implements several Java EE specifications including Java Servlet, JavaServer Pages(JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment in which Java code can run.

Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license, and is open-source software.

5.3 Code Discussion

/*Pseudocode Login Page */

Data: email, password

Result: Transfer of control to one of three user

```
if(check whether email is present in login)
{
    if(check whether password is correct) {
        if(email belong to student) {
            RequestDispatcher
            rd=request.getRequestDispatcher("student");

            rd.forward(request, response);
        }
        if(email belong to teacher)
        {
            RequestDispatcher
            rd=request.getRequestDispatcher("Teacher");

            rd.forward(request, response);
        }
        if(email belong to Parents) {
```



```
RequestDispatcher
rd=request.getRequestDispatcher("Parents");

rd.forward(request, response);

}else{

    print("Sorry Your Details are not Updated");

}else{

    print("Sorry incorrect Email id and password");

}

else

{

    print("Your Email Id is not in Login");

}}
```

/*Pseudocode Student page */

Data: usn, sem

Result: display the results of the student

Get the subject of a particular sem;

```
while(if there is next row of tuples) {

    if(there is first internal result) {

        print( the results of first internal);

    }

    if(there is second internal result) {

        print( the results second internal);

    }

    if(there is third internal result) {

        print( the results third internal);
```

```
}
```

```
if(all the three internal results are updated )
```

```
{
```

```
    print( the average internal marks);
```

```
}
```

```
}
```

```
/*Pseudocode teacher operation */
```

Data: operation to preform

Result: transfer to the page to perform the operation

```
if(to add a new student) {
```

```
    RequestDispatcher rs=getServletContext().getRequestDispatcher("/Studententry.html") ;
```

```
        rs.include(request, response);
```

```
    }
```

```
    if(to updated the internal results) {
```

```
        transfer the control to result update page;
```

```
    }
```

```
if(to connect parent to student) {
```

```
    RequestDispatcher rs=getServletContext().getRequestDispatcher("/parententry.html") ;
```

```
    rs.include(request, response);
```

```
    }
```

```
}
```

```
/*Pseudocode to add a new student */
```

Data: email, usn, name ,gender ,sem ,sec ,city ,phone;

Result: insertion of the new Student

If(email not present in login) {

 if(email not present in student) {

 create a connection to database;

 create a statement to execute the query ;

stmt.executeQuery("insert into login"+" values('"+email+"','CNS123') ");

stm.executeQuery("insert into
students"+"values('"+sid+"','"+name+"','"+gender+"','"+sem+"','"+sec+"','"+phone+"','"+city+"','"+ema
il+"') ");

stm.executeQuery("commit");

 }else{

 Print("Student is Successfully Entered");

 }

 }else{

 Print("Student is Already in Student table");

 }

/*Pseudocode to update a new results of the student */

Data: usn, internal ,marks of the internal;

Result: updation of the new results of the student

if(usn is present in Student table) {

 if(internal results not present in results) {

 create a connection to database;

 create a statement to execute the query ;

```
stm.executeQuery("insert into results"+  
values(""+sid+", "+inter+", "+mark1+", "+mark2+", "+mark3+", "+mark4+", "+mark5+", "+mark6+")  
");
```

```
stm.executeQuery("commit");
```

```
stm.close();
```

```
Print("Student Results is Successfully Entered");
```

```
}else{
```

```
Print("Students Internal marks is Updated");
```

```
}
```

```
}else{
```

```
Print("Students Details Doesn't Exist");
```

```
}
```

CHAPTER 6

DISCULUSION OF THE RESULTS

6.1 Code Discussion

/*Pseudocode Login Page */

Data: email, password

Result: Transfer of control to one of three user

if(check whether email is present in login)

{

if(check whether password is correct) {

if(email belong to student) {

RequestDispatcher rd=request.getRequestDispatcher("student");

rd.forward(request, response);

}

if(email belong to teacher)

{ RequestDispatcher rd=request.getRequestDispatcher("Teacher");

rd.forward(request, response);

}

if(email belong to Parents) {

RequestDispatcher rd=request.getRequestDispatcher("Parents");

rd.forward(request, response);

}else{

print("Sorry Your Details are not Updated");

}else{

print("Sorry incorrect Email id and password");

}else

{

print("Your Email Id is not in Login");

}}

/*Pseudocode Student page */

Data: usn, sem

Result: display the results of the student

Get the subject of a particular sem;

```
while(if there is next row of tuples) {  
    if(there is first internal result) {  
        print( the results of first internal);  
    }  
    if(there is second internal result) {  
        print( the results second internal);  
    }  
    if(there is third internal result) {  
        print( the results third internal);  
    }  
    if(all the three internal results are updated )  
    {  
        print( the average internal marks);  
    }  
}
```

/*Pseudocode teacher operation */

Data: operation to preform

Result: transfer to the page to perform the operation

```
if(to add a new student) {  
    RequestDispatcher  
    rs=getServletContext().getRequestDispatcher("/Studententry.html") ;  
    rs.include(request, response);  
}
```

```
if(to updated the internal results) {  
    transfer the control to result update page;  
}
```

```
if(to connect parent to student) {  
    RequestDispatcher rs=getServletContext().getRequestDispatcher("/parententry.html") ;  
    rs.include(request, response);  
}  
}
```

/*Pseudocode to add a new student */

Data: email, usn, name ,gender ,sem ,sec ,city ,phone;

Result: insertion of the new Student

```
If( email not present in login) {  
    if(email not present in student) {  
        create a connection to database;  
        create a statement to execute the query ;  
        stmt.executeQuery("insert into login"+" values('"+email+"','CNS123') ");  
        stm.executeQuery("insert into  
students"+"values('"+sid+"','"+name+"','"+gender+"','"+sem+"','"+sec+"','"+phone+"','"+city+"','"+  
email+"') ");  
        stm.executeQuery("commit");  
    }else{  
        Print("Student is Successfully Entered");  
    }  
}else{  
    Print("Student is Already in Student table");  
}
```

/*Pseudocode to update a new results of the student */

Data: usn, internal ,marks of the internal;

Result: updation of the new results of the student

if(usn is present in Student table) {

 if(internal results not present in results) {

 create a connection to database;

 create a statement to execute the query ;

 stm.executeQuery("insert into results"+"
values('"+sid+"','"+inter+"','"+mark1+"','"+mark2+"','"+mark3+"','"+mark4+"','"+mark5+"','"+mark6+"')
");

 stm.executeQuery("commit");

 stm.close();

 Print("Student Results is Successfully Entered");

 }else{

 Print("Students Internal marks is Updated");

 }

 }else{

 Print("Students Details Doesn't Exist");

 }

4.2 Snapshots

- Login Page where you enter Email id and password

CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

Welcome to the Login Page
Enter the Email Id and Password
Student Default password is CNS123

Enter the EMAIL to login

Enter the password

[Change Password](#)

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore, Karnataka || [HOME](#)

Fig 4.1 Login Page

- Page to Change Password

CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

Enter Your Email ID

Enter the Old Password

Enter the New Password

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore, Karnataka || [HOME](#)

Fig 4.2 Change password page

- Teacher page to select the option

CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

Welcome to Teacher Editorial Page Your Details

TID =	CNS49801	NAME =	CHETAN
Qualification =	BE,M-TEC	Phone =	7760683696
Address =	BANGALORE	Email Id =	CHETAN@GMAIL.COM

Select the Action You need to Perform

Insert the Student details ☐
 Insert the Results of the Students ☐
 Insert the Student to the Parents ☐

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[LOGOUT](#)

Fig 4.3 Teacher page

- To insert the Student details

CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

Enter Student Details

Enter the Student USN: Eg:1CN16IS000

Enter the Student Name:

Enter the Student Gender: MALE ☒ FEMALE ☐

Enter The Student SEM:

Enter the Student SEC: A ☒ B ☐

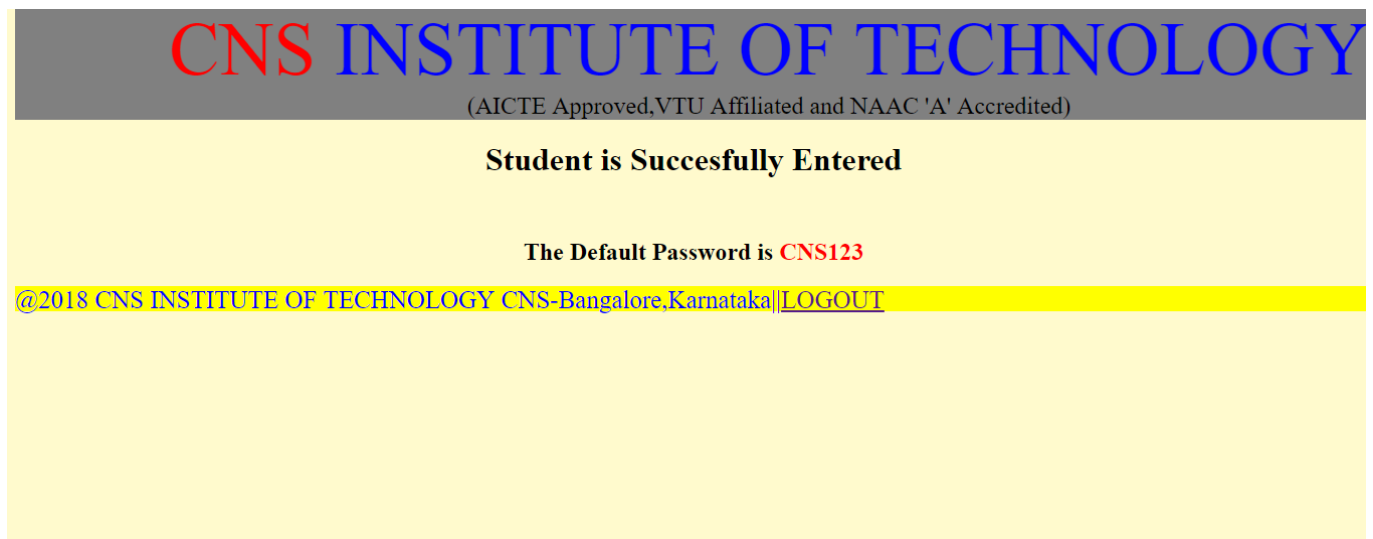
Enter the Student Phone Number:

Enter the Student City:

Enter the Student Email Id:

Fig 4.4 Student details entry page

- When Students details are Successfully Inserted



CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved,VTU Affiliated and NAAC 'A' Accredited)

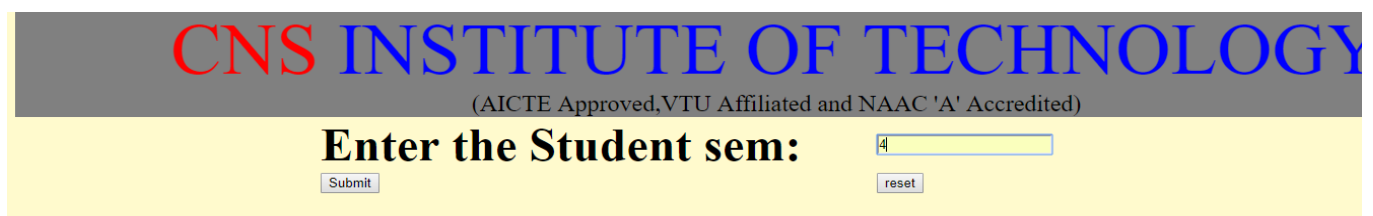
Student is Successfully Entered

The Default Password is **CNS123**

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[LOGOUT](#)

Fig 4.5 Students details are Successfully

- To select Semester of the Results to Insert

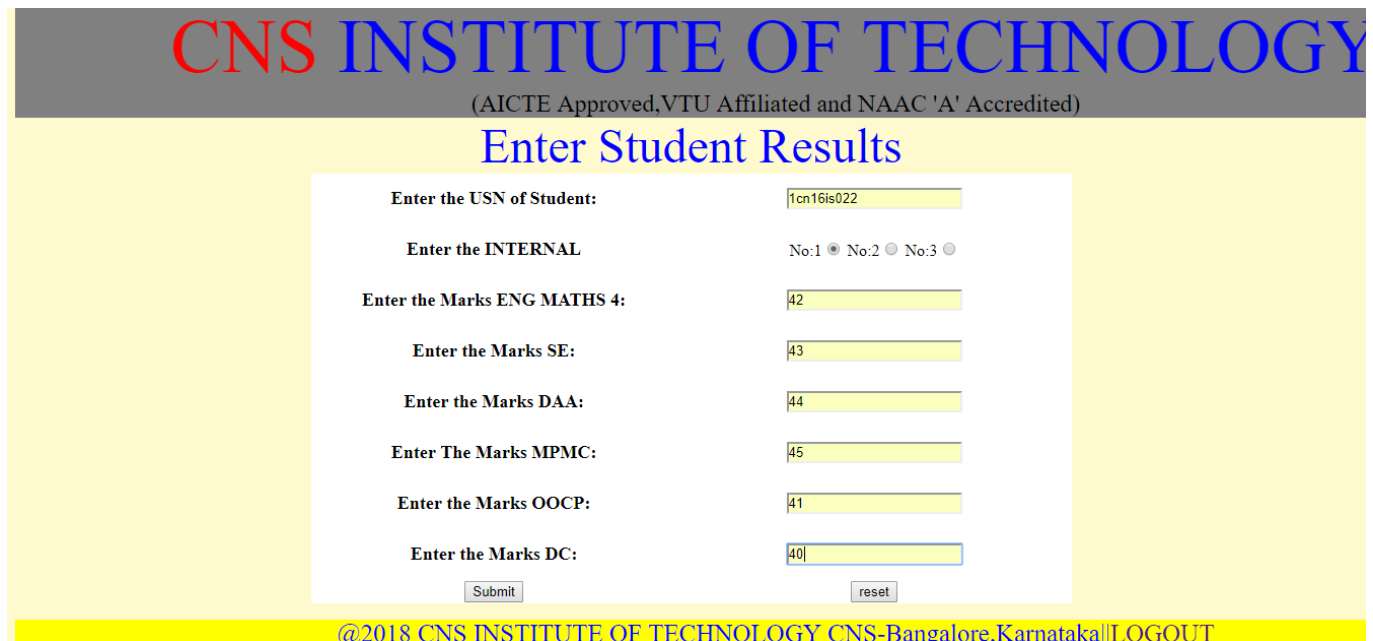


CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved,VTU Affiliated and NAAC 'A' Accredited)

Enter the Student sem:

Fig 4.6 Select Semester of the Results to Insert

- To insert the Results



CNS INSTITUTE OF TECHNOLOGY
(AICTE Approved,VTU Affiliated and NAAC 'A' Accredited)

Enter Student Results

Enter the USN of Student:

Enter the INTERNAL: No:1 ☒ No:2 ☐ No:3 ☐

Enter the Marks ENG MATHS 4:

Enter the Marks SE:

Enter the Marks DAA:

Enter The Marks MPMC:

Enter the Marks OOCp:

Enter the Marks DC:

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[LOGOUT](#)

Fig 4.7 Insert the Results

- When Results are Successfully Inserted

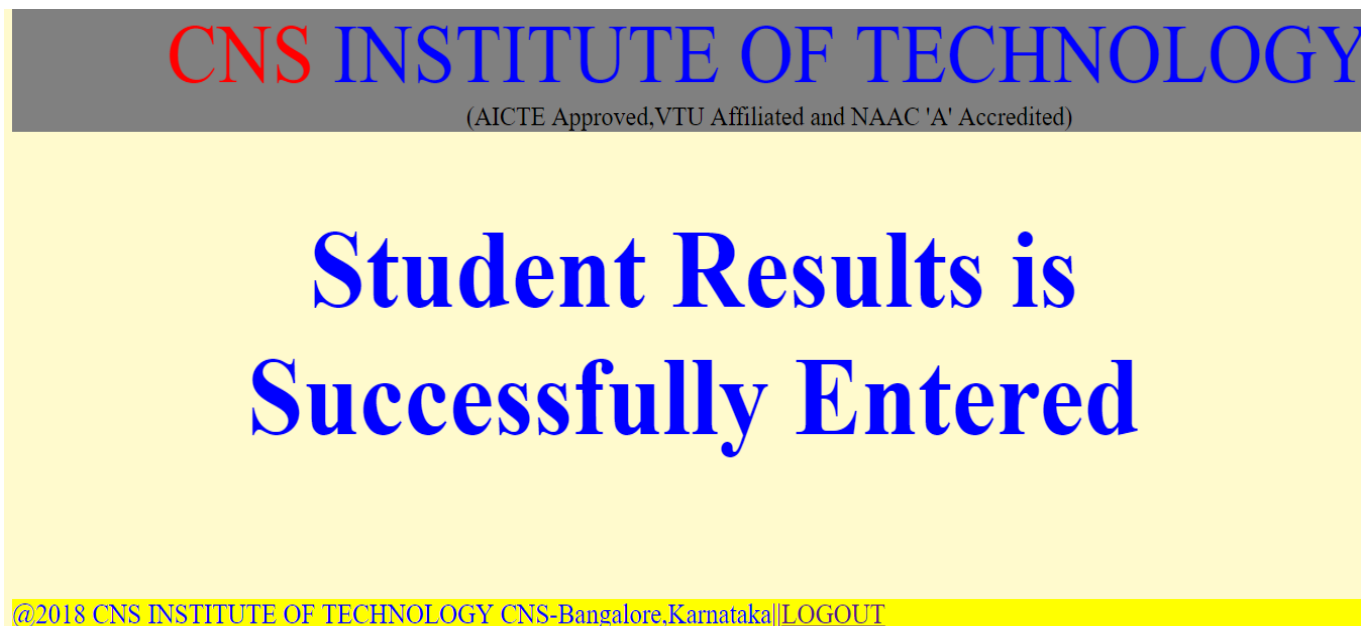


Fig 4.8 Results are Successfully Inserted

- To connect Parents and Students

The screenshot shows the header of the CNS Institute of Technology website. Below the header, there is a form with a white background. The form contains two labels: "Enter the EMAIL to Parent" and "Enter the USN of Child". The email field contains "mallika@gmail.com" and the USN field contains "1cn16is022". There are "Submit" and "Reset" buttons at the bottom of the form. Below the form, the text "To Repeat the Following Action || [Click Here!!!](#)" is displayed. At the bottom, a yellow footer bar contains the text "@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[HOME](#)".

Fig 4.9 To connect Parents and Students

- Page when the a Parent Login

CNS INSTITUTE OF TECHNOLOGY

(AICTE Approved,VTU Affiliated and NAAC 'A' Accredited)

Select One of The Child To See The Results

To View The Results of 1CN16IS022=●

To View The Results of 1CN16IS026=●

To View The Results of 1CN16IS027=●

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[LOGOUT](#)

Fig 4.10 Page when the a Parent Login

- Page when a Student Login

USN = 1CN16IS027
 Gender = M
 SEC = A
 Address = BANGALORE

NAME = MANU
 SEM = 5
 Phone = 7760683696
 Email Id = MANU@GMAIL.COM

INTERNAL				1	INTERNAL				2	INTERNAL				3
Subjects	Max Marks	Min Marks	obtained Marks		Subjects	Max Marks	Min Marks	obtained Marks		Subjects	Max Marks	Min Marks	obtained Marks	
ME	45	21	35		ME	45	21	39		ME	45	21	37	
CN	45	21	43		CN	45	21	40		CN	45	21	42	
ATC	45	21	40		ATC	45	21	42		ATC	45	21	45	
DBMS	45	21	42		DBMS	45	21	43		DBMS	45	21	42	
AI	45	21	30		AI	45	21	36		AI	45	21	45	
JAVA ADV	45	21	41		JAVA ADV	45	21	38		JAVA ADV	45	21	40	

Your Final Average Internal Marks

Subjects	Max Marks	Min Marks	obtained Marks
ME	20	12	16
CN	20	12	18
ATC	20	12	19
DBMS	20	12	14
AI	20	12	18
JAVA ADV	20	12	18

@2018 CNS INSTITUTE OF TECHNOLOGY CNS-Bangalore,Karnataka||[LOGOUT](#)

Fig 4.11 Page when a Student Login

Chapter 5

CONCLUSION AND FUTURE ENHANCEMENTS

The Examination Result Management System is developed using JAVA Servlet and ORACLE. It fully meets the objectives of the system which it has been developed. The system has reached a steady state where all bugs have been eliminated and also The system is operated at a high level of efficiency and all the teachers and students associated with the system understands its advantage. The system solves the problem. It was intended to solve as requirement specification.

The project can be implemented to send the notifications to student and parents as soon as the results are updated and to Shortlist students who don't have min average in internal. To notify the student about the upcoming internal test through the email. Update the Project to include the Attendance. Update the Project to include the all the Department.

REFERENCES

- [1] <https://www.w3schools.com/html/default.asp>
- [2] <https://www.javatpoint.com/java-swing>
- [3] <https://stackoverflow.com/questions/tagged/swing>
- [4] The Complete Reference Java(Seventh Edition) -Herbert Schildt