EMAIL INTERFACE SYSTEM

A PROJECT REPORT

Submitted By

Priya Garg – 2100290140107
Shivam Agrawal – 2100290140126
Prashant Sharma – 2100290140105

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
Dr. Vipin Kumar
Associate Professor

DEPARTMENT OF COMPUTER APPLICATIONS

KIET GROUP OF INSTITUTIONS, Ghaziabad

Uttar Pradesh-201206

(May 2023)

CERTIFICATE

Certified that **Prashant Sharma (2100290140105)**, **Shivam Agrawal (2100290140126), Priya Garg (2100290140107)** have carried out the project work **"Email Interface"** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the students themselves and the contents of the project report do not form the basis for the award of any other degree to the candidates or to anybody else from this or any other University/Institution

Prashant Sharma (2100290140105)

Shivam Agrawal (2100290140126)

Priya Garg (2100290140107)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar (Associate Professor)

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Arun Kumar Tripathi

Head, Department of Computer Applications

KIET GROUP OF INSTITUTIONS, GHAZIABAD

ABSTRACT

An Internet is a private computer network that uses internet protocol Technologies to securely share any one part of an organization information or network operating system with an that organization. Sometimes the term refer to the organization internal website but may be more extensive part of the organization and collaboration to compete, a company must find ways to communicate instantly and share knowledge across boundaries of time and distance that were once almost impossible to bridge.

“Intra mailing System” utilizes a wired range of immerging digital technology to bridge this gap and give the people in an organization real time access to the information they need no matter where they are located. “Intra Mailing System” provide fast, easy and secure excess tools for communicating with in the organization network throw mail, text, images and instant messages .

Intra Mailing System aims at connecting all the people in an organization and with in its multiple branches.

ACKNOWLEDGMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, Dr. Vipin Kumar for his guidance, help and encouragement and throughout my research work. Their en-lightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor Head, Department of Computer Applications, for his insightful comments And administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thank go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep life filled with enjoyment happiness.

PRIYA GARG (University Roll no. 2100290140107)

PRASHANT SHARMA (University Roll no. 2100290140105)

SHIVAM AGRAWAL (University Roll no. 2100290140126)

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

A company has different departments situated in different locations. One as four office and three sites (factory). These are situated at different places and have a minimum distance The basic problem is that they communicate or make correspondence only through post as their units are located in remote areas. This method of communication is very slow in nature and it’s not reliable.

Therefore, the solution for this problem is to develop a communication channel that meet the need of the organization. Basically, the computerized solution is for improving the services as well as self-efficiency of the computer.

This project “Intra Mailing System” aims at providing a good communication interface for the organization. It can cut down the time of employees send on routine communication tasks. Its successful Implementation makes easy the job of employees in the organization.

This application provides facilities like mails, instant message and allows employees to communicate more effectively, and access the resources they need to maximize their day-to-day productivity. It provides fast and better access to up-to-date information.

“Intra Mailing System” is the system which is used to establish online communication among the employees of an organization at different branches and within the organization between the departments. The employees of the organization can send mails to each other in different departments at different locations.

The purpose of “Intra Mailing System” is, an employee of a company can send and receive mails only in that organization’s Local Area Network or Wide Area Network. It will worklike intranet mailing system for the internet.

1.2 PROJECT DESCRIPTION

The project is desired to meet the following objectives:

The objective of this software is to completely automate the process of organization. User can send information and query to each other. Provide the facilities that improve the procedures within organizations. To maintain and store the information about the process of organization. Provide excellent and easy to access means of communication medium between Employees and Admin.

Provide Classifieds as a service to enable its Employees to connect with one a separately.

EXISTING PROBLEM

The manual work and messge which is nesaccery to convey to from one user to other can be manage easily and rapidly.

Like:

- Sending some information via mail.
- Communication of the user is now easy by using chat in the network.
- Sending attachment of file via mail

1.3 PROJECT SCOPE

One of the objectives behind developing this software (Intranet Mail System) is to fulfill all the communication needs of the employees as well as the clients in an organization in a user friendly and efficient manner.

This would be a web-based application which provides a single platform for all the employees or users of an organization sitting at different locations to intranet with each other and make a community for messaging, blogs, forums & reports and query. It helps in making the communication easy and faster.

Another objective is to manage all the departments, roles, employees and all the queries of the user. Here different employees of the organization can easily send messages to each other and interact. They can even upload their daily activity reports, weekly activity reports and monthly activity reports.

1.4 HARDWARE/ SOFTWARE USED IN THE PROJECT

1.4.1 HARDWARE REQUIRMENT FOR DEVELOPMENT –

FRONT END

Java Script/Servlet

- User friendly
- GUI
- Separation of work (designing & coding)
- Written once run anywhere
- Java Beans

BACK END

Postgress

- Security
- Performance
- Scalability
- Reliability
- Support RDMS concepts

HARDWARE REQUIRMENTS

❖ Processor	Intel i5
❖ RAM	4GB

Hard Disk	1SSD
Display	HD(1200X1000 Resolution)

Software:-

Operating System	Windows 11
Front End	JAVA SCRIPT/SERVLET/JSP.
Back end	POSTGRESS

CHAPTER 2

FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. This type of study determines if a project can and should be taken. A feasibility study is carried out to select the best system that meets performance requirements. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability or meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development. The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility.

To check whether the proposed system is worth making or not I conducted the feasibility study in which I studied the following aspects.

2.1 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the sytem requirement in the terms of input, outout, program and procedure. Having identified an outline system, the investigation must go to suggest the type of equipment, required method developing the sytem, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?

 Can the sytem expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete ofter some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using java the project is technically feasible for development.

Here we are concerned with specifying the equipments and software that will successfully satisfy the requirements of the system. While studying the technical feasibility I observed the following aspects:

- There is adequate and regular power supply.
- There is easy availability of intranet network and also client of the system can avail the connection with minimum cost.
- Specified H/W like processor of about 1GHz with a 256MB RAM and

S/W like JAVA and MySql are already available with the client.

1. Tools Used:
1. J2SDK 6.0
2. JDK 1.6
3. weblogic sever
4. Postgress

After observing the above said aspects I can say that this system is technically feasible.

2.1 OPERATIONAL FEASIBILITY

Here we consider the human aspect of the organization. This test of feasibility asks if the system will work when it is developed and installed. In this stage I observed the aspects.

- Everyone welcomes the new system and there was hardly any resistance because of ease of access and user friendly environment.
- The existing staff is skilled enough and is computer literate to handle the new system.
- It was perceived that the system, when launched, will do no harm to the existing business.

After observing the above aspects it was concluded that the system is operationally feasible.

2.2 BEHAVIOURAL FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. Here we are concerned with the economical aspect of overall system. A system that can be developed technically and that will be used if installed, must still be profitable for the organization. Here I estimated the following costs:

- One time development cost
- One time H/W and S/W cost
- Periodic maintenance cost

After analysis it was found that this cost was less than the current system cost.

It also provides the following features:

- Ease of access
- Fast processing
- Fast retrieval of information

After these studies it was concluded that this system is economically feasible.

CHAPTER 3

DATABASE DESIGN

Database design transform the information domain model created during analysis into the data structure that will requires implementing the software.

3.1 COMPOSE TABLE

This Table 3.1 Compose Table Contains the following attributes Mail id, Mail from, Mail to, Mail Subject, Mail Bcc, Mail Date, Folder with the primary key mail id and which helps to send the data at the correct destination.

Field name	**Null?**	**Data type**
MailId	NOT NULL	Int
Mailfrom		Varchar(50)
Mailto		Varchar(50)
Subject		Varchar(80)
Mailcc		Varchar(50)
MailBcc		Varchar(50)
MailData		Varchar(99)
Folder		Varchar(50)
MailDate		Varchar(35)
MailSt		Int

Table 3.1: Compose Table

3.2 SIGN UP TABLE

This Table 3.2 Sign Up Table contains the attributes Uname, Password, Age, Sex, City, State, Pin, Country with primary key Uname which help us to enter the details of new user in the database.

Field Name	**Null?**	**Data Type**
Uname	**NOT NULL**	**Varchar(50)**
Passwd		**Varchar(30)**
Age		**Int**
Sex		**Varchar(6)**
City		**Varchar(50)**
State		**Varchar(50)**
Pin		**Varchar(20)**
Country		**Varchar(40)**

Table 3.2: Sign Up Table

3.3 NEW FOLDER

This Table 3.3 Contains the attributes Mid, Mfrom, Mto, Sub, Mcc, Mdata, Folder, MailSt, MailDate with the primary key Mail id which can maintain the link to maintain the address table.

Field Name	Null?	Data Type
MID	NOT NULL	Int
MFrom		Varchar(40)
MTo		Varchar(40)
Sub		Varchar(40)
Mcc		Varchar(40)
MData		Long
Folder		Varchar(15)
MailSt		Int
MailDate		Varchar(35)

Table 3.3: New Folder

3.4 ADDRESS TABLE

This Table contains the attributes ActName, Uname, NickName, Email Id, Addresses, Phone with the primary datatype varchar that maintains the address of banner advertiser.

Field Name	**Null?**	**Data Type**
ActName		**Varchar(50)**
Uname		**Varchar(50)**
NickName		**Varchar(50)**
EmailId		**Varchar(50)**
Addresses		**Varchar(99)**
Phone		**Int**

Table 3.4 Address 1

3.5 CHAT MESSAGE TABLE

This Table Contains the attribute Username, Message, Date, Response_id, Response_text with Null constraints Field Name Attribute and not null constraint username attribute which can maintain the record of message

Field Name	**Null?**	**Data Type**
Username	**NOT NULL**	**Varchar(40)**
Message		**Varchar(40)**
Date		**Varchar(30)**
Response_id		**Int**
Response_text		**Varchar(99)**

Table 3.5 Chat Message

.

3.6 SEARCH TABLE

This Table contains the Attributes Mid, Mail Data, Data of view with the constraints Null Field name attribute and not null mail id attribute which helps to search the data in the search table.

Field Name	Null?	Data Type
Mid	NOT NULL	Int
MailData		Varchar(99)
Date of view		Varchar(40)

Table 3.6 Search Table

3.7 CATEGORY TABLE

This table contains the attribute Category_id, Category_name, Category_description with the constraint Null Field name and not null category_id which helps to categorize the mail.

Field Name	**Null?**	**Data Type**
Category_id	**NOT NULL**	**Int**
Category_name		**Varchar(50)**
Category_description		**Varchar(99)**

Table 3.7 Category Table

3.8 LOGIN TABLE

This table contains the attributes Username and password which is for store the details of the user with the constraints not null field name username .

Field Name	**Null?**	**Data Type**
Username	**NOT NULL**	**Varchar(40)**
Password		**Varchar(40)**

Table 3.8 Login

3.9 FLOW CHART

In this data flow diagram (3.2.1) shows the flow of data first of all user can register on the Intranet Mail System with following details Name, age, sex, state, city, pincode, id. Then user can login with the help of details on the internet mailing system and it can allow to the user to send the mail to the specified receiver and admin can control all over the internet mailing system like they can search or view the mail of any user.

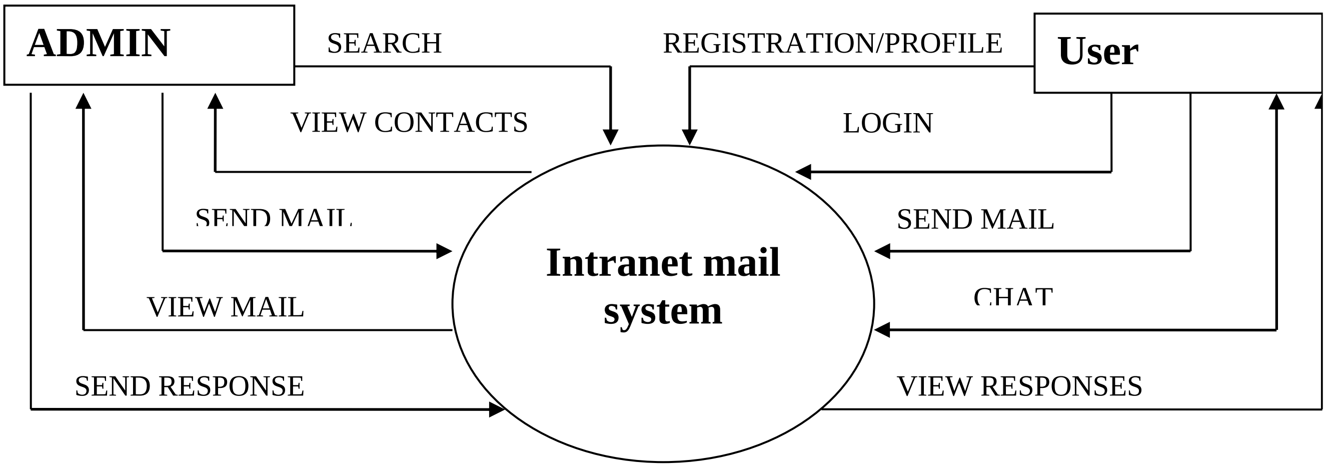


Figure 3.2: DFD

CHAPTER 4

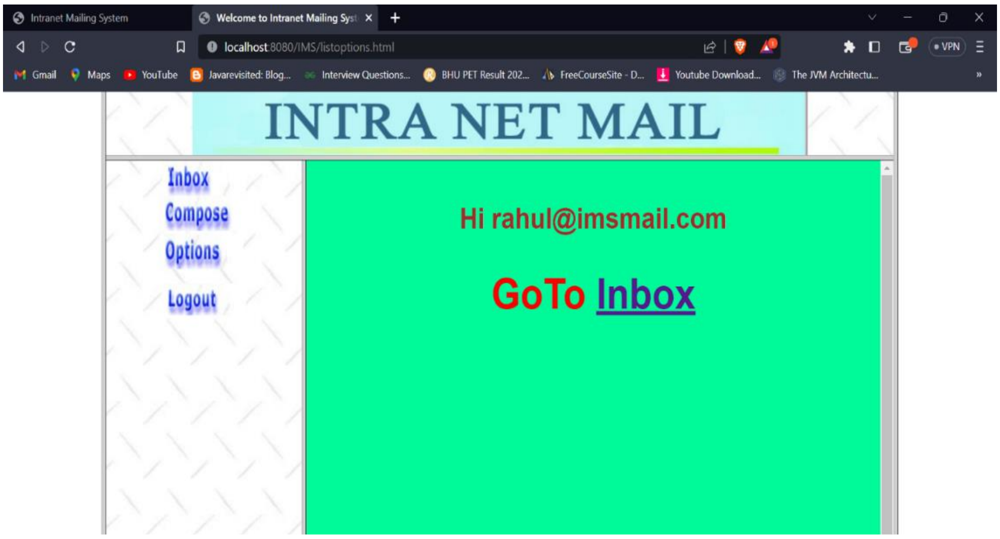
FORM DESIGN

4.1 INPUT/OUTPUT (SCREENSHOT)

In this chapter we show the input output form i.e the how the user can take intract with gui how user can send the mail and how the mail is going to the server by using the input of the user details.

4.1 LOGIN IMAGE

In Figure 4.1.1 shows that when the user can login to the internet mailing system they show the message HII user@ims mail.com and show the 4 options



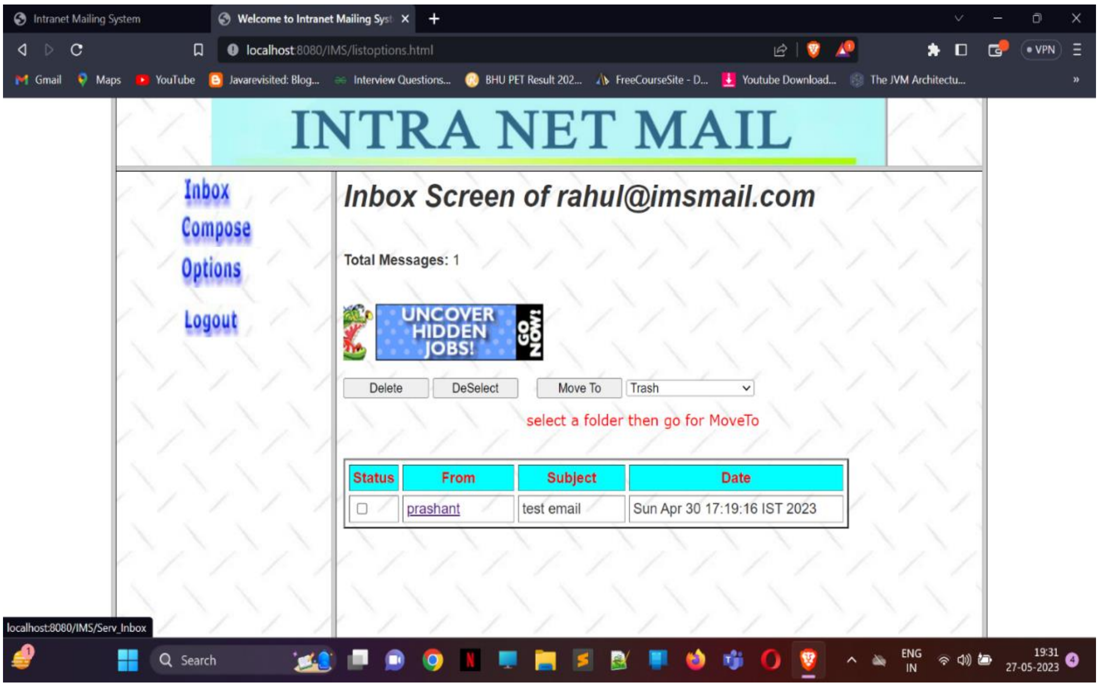
Hi rahul@imsmail.com

GoTo Inbox

Figure 4.1: Inbox

4.2 INBOX IMAGE

In figure 4.1.2 shows the module inbox view when the user can enter in the inbox then they can see there mails with respective date, subject and details of sender.



Inbox Screen of rahul@imsmail.com

Total Messages: 1

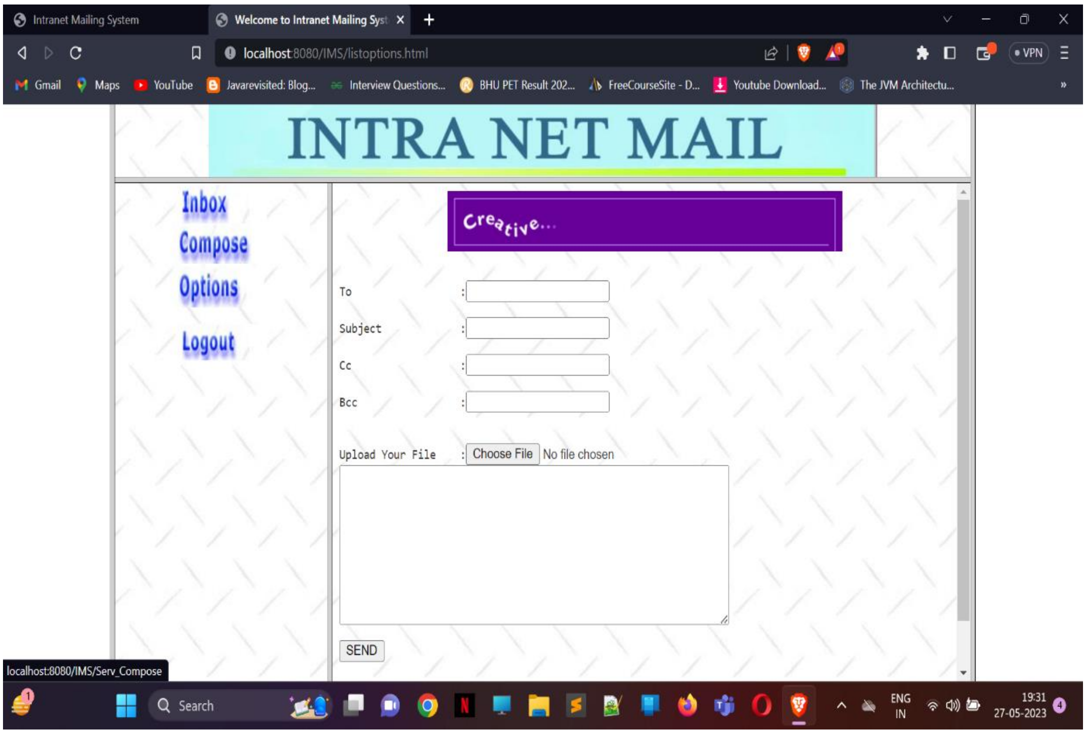
select a folder then go for MoveTo

Status	From	Subject	Date
□	prashant	test email	Sun Apr 30 17:19:16 IST 2023

Figure 4.2: Inbox

4.3 COMPOSE IMAGE

In figure 4.3 shows the module compose . In this module user can enter the details of the receiver where they can send the mail with respective subject and also user can attach any file to the receiver.


Intranet Mailing System
Welcome to Intranet Mailing Syst
localhost:8080/IMS/listoptions.html
INTRA NET MAIL
Inbox
Compose
Options
Logout
Creative...
localhost:8080/IMS/Serv_Compose

To :

Subject :

Cc :

Bcc :

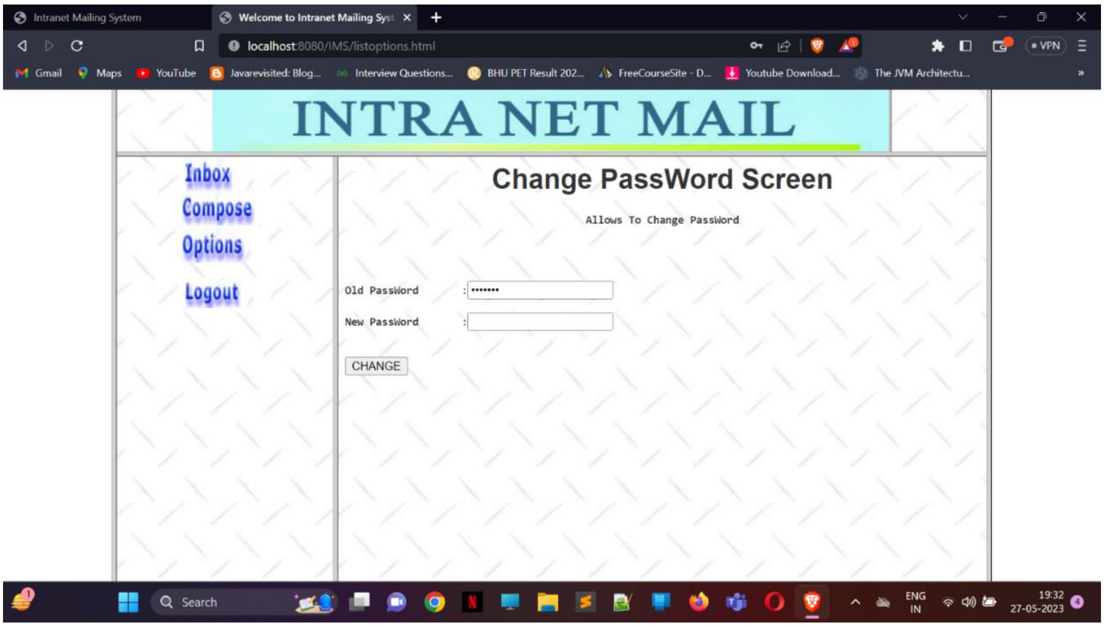
Upload Your File : Choose File No file chosen

SEND

Figure 4.3: Compose

4.4 OPTIONS IMAGE

In this figure 4.1.4 module option provide two option old password and new password which can helps us to user to change the password .



INTRA NET MAIL

Change PassWord Screen

Allows To Change PassWord

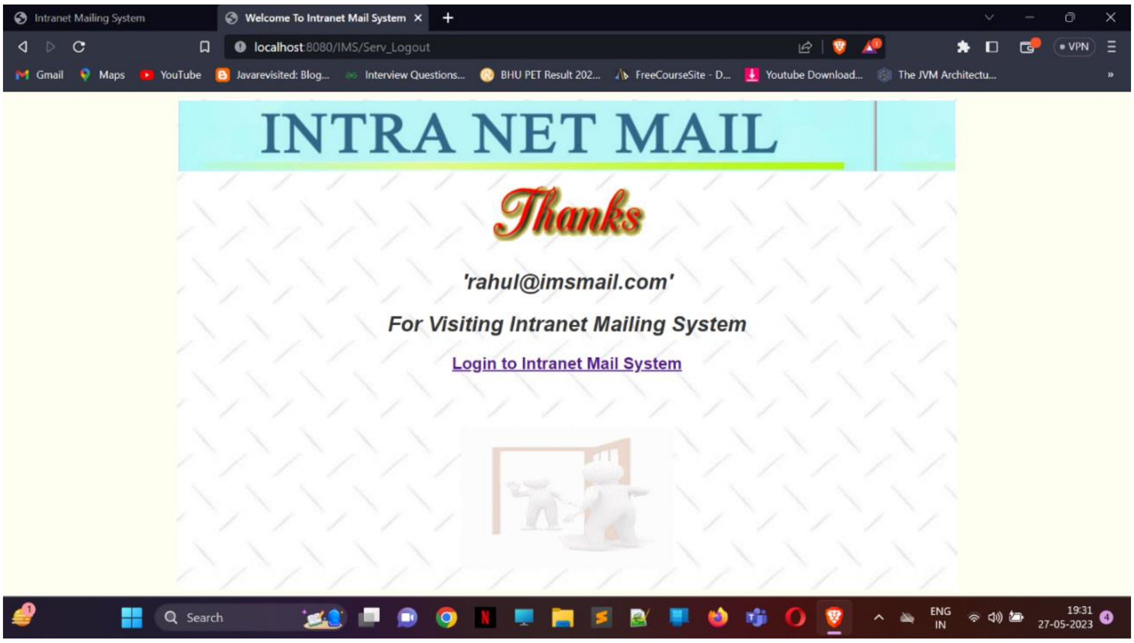
Old PassWord : •••••••

New PassWord :

Figure 4.4: Option

4.5 LOGOUT IMAGE

In this Figure 4.1.5 shows the module logout where user can logout the account from the internet mailing system.



INTRA NET MAIL

Thanks

'rahul@imsmail.com'

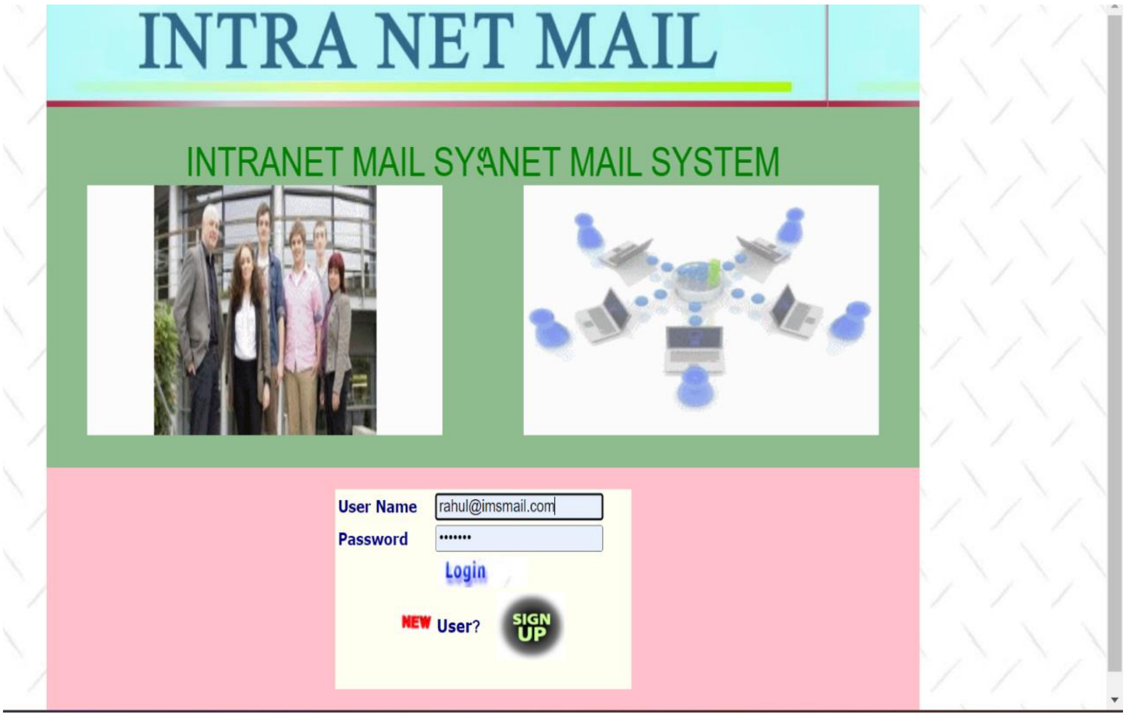
For Visiting Intranet Mailing System

Login to Intranet Mail System

Figure 4.5: Logout

4.6 LOGIN IMAGE

In this Figure 4.6 after the sign up user can login in the internet mailing system with the following details username and password and easily access there account.



User Name rahul@imsmail.com

Password •••••••

Figure 4.6: Login

CHAPTER 5

CODING

5.1 Sign Up

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.ServletException;

import javax.servlet.ServletOutputStream;

import javax.servlet.http.*;

import connect.ConnectionProvider;

public class Serv_SignUp extends HttpServlet

{

   public Serv_SignUp()

   {

   }

   public void service(HttpServletRequest httpservletrequest,
HttpServletResponse httpservletresponse)

       throws ServletException, IOException

   {

       try

       {
        //Class.forName("oracle.jdbc.driver.OracleDriver");

        //con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system",
"ojasvi");


con=ConnectionProvider.getConnection();

        st = con.createStatement();

        ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

        p = httpservletrequest.getParameter("pwd1");

        HttpSession httpsession = httpservletrequest.getSession(true);

        httpsession.putValue("name",
httpservletrequest.getParameter("uname1"));

        u = httpsession.getValue("name").toString();

        rs = st.executeQuery("select * from signupdetails where uname='" + u +
"' and  passwd='" + p + "'");

        if(rs.next())

        {

           httpservletresponse.sendRedirect("listoptions.html");

        } else

        {

           servletoutputstream.println("<html><body bgcolor=white
background='INDTEXTB.JPG' text=red><h2><i><b>ur not a valid user! Try
again using correct Loginname & Password or try registering
</b></i></h2></body></html>");
```

```
                httpservletresponse.setHeader("Refresh", "3;URL=loginsc.html");
            }
        }
        catch(Exception exception)
        {
            exception.printStackTrace();
    }
    }
    Connection con;
    Statement st;
    static String u;
    static String p;
    ResultSet rs;
}
```

5.2 Sent Message

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;


import connect.ConnectionProvider;


public class Serv_SentMessages extends HttpServlet

{


    public Serv_SentMessages()

    {

        st = null;

        con = null;

        scon = null;

        rs = null;

        s = "";

        d = "";

        mt = "";

        mcc = "";

        col = 1;

    }
```

```
   public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)
        throws ServletException, IOException
    {
        try
        {
         //   Class.forName("oracle.jdbc.driver.OracleDriver");
         //   con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system", "ojasvi");

            con=ConnectionProvider.getConnection();

            st = con.createStatement();
            scon = getServletContext();
            HttpSession httpsession = httpservletrequest.getSession(true);
            uname = httpsession.getValue("name").toString();
            ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();
            servletoutputstream.println("<html><head><title>Welcome To Intranet Mailing
System</title>");
            servletoutputstream.println("<script>function d1(){");
            servletoutputstream.println("document.ff.action='Serv_DeleteMessage';");
            servletoutputstream.println("document.ff.submit();");
            servletoutputstream.println("}</script></head>");
        servletoutputstream.println("<body bgcolor=white
background='INDTEXTB.JPG' text=blue><h3><i>Welcome '" + uname + "' @Intranet
Mailing System</i></h3>");

        servletoutputstream.println("<form name=ff><input type=button name=del
value='Delete' onClick='d1()'> <input type=reset name=des value='DeSelect'>");

        servletoutputstream.println("<h3><i>Messages sent by " + uname + " till
now</i></h3><br><br>");

        servletoutputstream.println("<table border=2 cellpadding=4 cellspacing=5
width=80%><tr
bgcolor=brown><th>X</th><th>To</th><th>Subject</th><th>Date</th></tr>");

        rs = st.executeQuery("select mailid,mailto,subject,mailcc,maildate from
newcompose where mailfrom='" + uname + "'");

        int i = 0;

        for(; rs.next(); servletoutputstream.println("<tr><td><input type=checkbox
name=cb" + i + " value=" + m + "></td><td><a href='Serv_SeeMessage1?id=" + m +
"'>" + mt + "</a></td><td>" + s + "</td><td>" + d + "</td></tr>"))

        {

          i++;

          m = rs.getInt(1);

          mt = rs.getString(2);

          s = rs.getString(3);

          mcc = rs.getString(4);

          d = rs.getString(5);

          if(col % 2 == 0)

            color = "pink";

          else

            color = "violet";
            bgcolor = "+color+";
            col++;
        }

        servletoutputstream.println("<input type=hidden name=h value=" + i + ">");
        servletoutputstream.println("</table></form></body></html>");
    }
    catch(Exception exception)
    {
        exception.printStackTrace();
    }
}

Statement st;
Connection con;
ServletContext scon;
ResultSet rs;
String uname;
String s;
String d;
String mt;
String mcc;
int m;
int col;
```

```
    String color;

    String bgcolor;

}
```

5.3 Compose Message

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;


import connect.ConnectionProvider;


public class Serv_Compose extends HttpServlet

{


    public Serv_Compose()

    {

        con = null;

        st = null;

        scon = null;

        rs = null;

        sos = null;

    }


    public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

        throws ServletException, IOException

    {
        try
        {
    //      Class.forName("oracle.jdbc.driver.OracleDriver");
    //      con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "ojasvi");


            con=ConnectionProvider.getConnection();


            st = con.createStatement();
            scon = getServletContext();
            String s = null;
            HttpSession httpsession = httpservletrequest.getSession(true);
            s = httpsession.getValue("name").toString();
            ServletOutputStream servletoutputstream = httpservletresponse.getOutputStream();
            servletoutputstream.println("<html><title>Intranet Mailing System</title>");
            servletoutputstream.println("<body bgcolor=white background='INDTEXTB.JPG' text=blue><center><img src='Images/ADV3.GIF' border=0></center>");
            servletoutputstream.println("<pre>");
            servletoutputstream.println("<form name=form4 method=post action='Serv_Send' enctype='multipart/form-data'>");
            servletoutputstream.println("To              :<input type=text name=to value= ''><br>");
            servletoutputstream.println("Subject         :<input type=text name=subject value=''><br>");
            servletoutputstream.println("Cc                :<input type=text name=cc
    value="><br>");
            servletoutputstream.println("Bcc               :<input type=text name=bcc
    value="><br><br>");
            servletoutputstream.println("Upload Your File    :<input type='file'
    name='file'/>");


            servletoutputstream.println("<textarea name=maildata rows=10
    cols=60></textarea><br>");
            servletoutputstream.println("<input type=submit name=submit1
    value='SEND'><br>");
            rs = st.executeQuery("select actname,emailid from address where uname='" + s
    + "'");
            if(rs.next())
            {
              servletoutputstream.println("<SELECT name=se onclick=getName(value)
    style=\"LEFT: 400px; POSITION: absolute; TOP: 160px; BACKGROUND-
    COLOR:#b464ff;width:130px;FONT-WEIGHT:BOLD\"");
              servletoutputstream.println("SIZE=5 name=List1 value=\"List1\">");
              do
                servletoutputstream.println("<OPTION value=" + rs.getString(2) + " >" +
    rs.getString(1) + "</OPTION>");
              while(rs.next());
              servletoutputstream.println("</SELECT>");
            }
            servletoutputstream.println("</form></body></html>");
            servletoutputstream.println("<script language=javascript>");
```

```
            servletoutputstream.println("function getName(Myname){");

            servletoutputstream.println("document.form4.to.value=Myname");

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

        }

        catch(Exception exception)

        {

            exception.printStackTrace();

        }

    }


    Connection con;

    Statement st;

    ServletContext scon;

    ResultSet rs;

    ServletOutputStream sos;

}
```

5.4 Change Password

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;

import connect.ConnectionProvider;

public class Serv_ChangePwd extends HttpServlet

{

public Serv_ChangePwd()

   {

   }



   public void init(ServletConfig servletconfig)

   {

      try

      {

         super.init(servletconfig);

    //      Class.forName("oracle.jdbc.driver.OracleDriver");

     //     con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system", "ojasvi");
```

```
            con=ConnectionProvider.getConnection();

            st = con.createStatement();

            scon = getServletContext();

        }

        catch(Exception exception)

        {

            exception.printStackTrace();

        }

    }

    public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

        throws ServletException, IOException

    {

        try

        {

            HttpSession httpsession = httpservletrequest.getSession(true);

            un = httpsession.getValue("name").toString();

            ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

            String s = httpservletrequest.getParameter("np");
            int i = st.executeUpdate("Update signupdetails set passwd='" + s + "' where uname='" + un + "'");

            servletoutputstream.println("<html><body bgcolor=white background='INDTEXTB.JPG' text=blue><h2><i>password is changed</i></h2></body></html>");

            httpservletresponse.setHeader("Refresh", "2;URL=Serv_Inbox");

        }

        catch(Exception exception)

        {

            exception.printStackTrace();

        }

    }



    Statement st;

    Connection con;

    ResultSet rs;

    ServletContext scon;

    String un;

}
```

5.5 Logout

```
import java.io.IOException;
import javax.servlet.*;
import javax.servlet.http.*;

public class Serv_Logout extends
HttpServlet
{

   public Serv_Logout()
   {
   }

   public void
service(HttpServletRequest
httpservletrequest,
HttpServletResponse
httpservletresponse)
      throws ServletException,
IOException
   {
      try
      {
         scon = getServletContext();
         HttpSession httpsession =
httpservletrequest.getSession(true);
         name =
httpsession.getValue("name").toString
();

httpsession.removeValue("name");
      }
      catch(Exception exception)
```

```
        {
           exception.printStackTrace();
        }
        ServletOutputStream
servletoutputstream =
httpservletresponse.getOutputStream()
;

servletoutputstream.println("<html><t
itle>Welcome To Intranet Mail
System</title>");

servletoutputstream.println("<body
bgcolor='Ivory' text=blue><center>");



servletoutputstream.println("<table
background='INDTEXTB.JPG'
width='70%' height='100%'>");

servletoutputstream.println("<tr><td
background='Images/IMS.JPG'
width='100%'
height='80'></td></td>");

servletoutputstream.println("<tr><td>
<center><img
src='Images/THANKS.GIF'></center>
</td><tr>");

servletoutputstream.println("<tr><td
        align='center'><h2><i>'" + name +
        "'</i></h2>");

        servletoutputstream.println("<h2><i>
        For Visiting Intranet Mailing
        System</i></h2>");
            servletoutputstream.println("<a
        href='loginsc.jsp'
        target=_parent><h3>Login to Intranet
        Mail System</h3></a></td></tr>");

        servletoutputstream.println("<tr><td
        align='center'><img
        src='Images/logout.gif' width='240'
        height='160'></td></tr>");

        servletoutputstream.println("</table>"
        );



        servletoutputstream.println("</body><
        /html>");
          }

          ServletContext scon;
          String name;
}
```

5.6 INBOX

```
import java.io.IOException;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

import connect.ConnectionProvider;

public class Serv_Inbox extends HttpServlet
{

    public Serv_Inbox()
    {
        st = null;
        st1 = null;
        con = null;
        scon = null;
        rs = null;
        rs1 = null;
        newfol = null;
        f1 = null;
        uname = null;
        sub = null;
        from = null;
        date1 = null;
    }

    public void service(HttpServletRequest
httpservletrequest, HttpServletResponse
httpservletresponse)
        throws ServletException, IOException
    {
        try
        {



con=ConnectionProvider.getConnection();

            st = con.createStatement();
            scon = getServletContext();
            HttpSession httpsession =
httpservletrequest.getSession(true);
            uname =
httpsession.getValue("name").toString();
            ServletOutputStream
servletoutputstream =
httpservletresponse.getOutputStream();

servletoutputstream.println("<html><head><t
itle>Welcome To Intranet Mailing
System</title>");

servletoutputstream.println("<script>function
d()");

servletoutputstream.println("{document.f.acti
on='Serv_DeleteCkd';");

servletoutputstream.println("document.f.subm
it();}");
            servletoutputstream.println("function
e()");
```

```
servletoutputstream.println("{document.f.acti
on='Serv_Enter';");

servletoutputstream.println("document.f.subm
it();}");

servletoutputstream.println("</script></head>
");
            servletoutputstream.println("<body
bgcolor=white
background='INDTEXTB.JPG' text=blue>");

servletoutputstream.println("<blink><h1><b>
<i>Inbox Screen of " + uname + "
</i></b></h1></blink><br>");
            int i = 0;
            for(ResultSet resultset =
st.executeQuery("select mailid from
newcompose where mailto='" + uname + "'
and folder='inbox'"); resultset.next();)
                i++;

            servletoutputstream.println("<b>Total
Messages:</b> " + i + "<br><br>");
            servletoutputstream.println("<br><a
href='http://www.monsterindia.com'
target=new><img src='jobsearch1.gif'
border=0></a><br><br>");
            servletoutputstream.println("<form
name=f action='Serv_MoveFolder'>");
```

```
            servletoutputstream.println("<input
type=button name=delete value='Delete'
onClick='d()' style=\"width:100\">");
            servletoutputstream.println("<input
type=reset name=deselect value='DeSelect'
style=\"width:100\">");

servletoutputstream.println("&nbsp;&nbsp;&
nbsp;&nbsp;<input type=submit name=move
value='Move To' style=\"width:100\">");
            ResultSet resultset1 =
st.executeQuery("select folder from folders
where uname='" + uname + "'");
            servletoutputstream.println("<select
name=se style=\"width:150\">");
            for(; resultset1.next();
servletoutputstream.println("<option value="
+ f1 + ">" + f1 + "</option>"))
                f1 = resultset1.getString(1);



servletoutputstream.println("</select><br>");
            servletoutputstream.println("<pre>
<font face=verdana color=red>select a folder
then go for MoveTo</font></pre>");
            resultset1.close();
            st.close();

servletoutputstream.println("<br><table
border=2 cellpadding=4 cellspacing=5
width=80%><TR
bgcolor=#00FFFF><th><font
color=red>Status</th><TH><font
```

```
color=red>From</TH><TH><font
color=red>Subject</Th><TH><font
color=red>Date</TH></TR>");
            st1 = con.createStatement();
            rs1 = st1.executeQuery("select
mailid,mailfrom,subject,maildate,mailst from
newcompose where mailto='" + uname + "'");
            int j = 0;
            while(rs1.next())
            {
                j++;
                m = rs1.getInt(1);
                from = rs1.getString(2);
                sub = rs1.getString(3);
                date1 = rs1.getString(4);
                sta = rs1.getInt(5);
                if(sub == null)
                    sub = "[NONE]";
                if(sta == 1)
                {

servletoutputstream.println("<tr><td><center
><img src='button.gif' height=20 width=25
></td><td>");

servletoutputstream.println("<input
type=checkbox name=b" + j + " value=" + m
+ "></td><td><a
href='Serv_SeeMessage?id=" + m + "'>" +
from + "</td></a><td>" + sub + "</td><td>"
+ date1 + "</td></tr>");
                } else
                {
```

```
servletoutputstream.println("<tr><td
width=6%><input type=checkbox name=b" +
j + " value=" + m + "></td><td
width=24%><a href='Serv_SeeMessage?id="
+ m + "'>" + from + "</td></a><td
width=23%>" + sub + "</td><td
width=47%>" + date1 + "</td></tr>");
        }
      }
      servletoutputstream.println("<script
language=javascript>");
      servletoutputstream.println("function
chkit(){");
      for(int k = 1; k <= j; k++)

servletoutputstream.println("document.f.b" +
k + ".checked=document.f.chkall.checked;");



servletoutputstream.println("}</script>");
      servletoutputstream.println("<input
type=hidden name=hid  value=" + j + ">");

servletoutputstream.println("</form>");

servletoutputstream.println("</table></body>
</html>");
    }
    catch(Exception exception)
    {
      exception.printStackTrace();
    }
    }

    Statement st;
    Statement st1;
    Connection con;
    ServletContext scon;
    ResultSet rs;
    ResultSet rs1;
    String newfol;
    String f1;
    String uname;
    String sub;
    String from;
    String date1;
    int sta;
    int m;
}
```

5.7 DOWNLOAD MESSAGE

```
import javax.servlet.*;

import java.io.*;

import javax.servlet.http.*;

public class Serv_Download extends HttpServlet

{

/**

         *

         */

        private static final long serialVersionUID = 1L;


public void service(HttpServletRequest request,HttpServletResponse response)throws
ServletException,IOException

{




OutputStream out=response.getOutputStream();

String downPath=this.getServletContext().getRealPath("attachment");


String name=request.getParameter("from");
response.setHeader("content-disposition","attachment;filename='"+name+"'");

response.setContentType("aplication/octet-stream");



downPath=downPath+"\\"+name;



System.out.println("path from where file to be downloaded.........."+downPath);




FileInputStream fin=new FileInputStream(downPath);

byte b[]=new byte[fin.available()];

fin.read(b);

out.write(b);

out.close();

fin.close();

}

}
```

5.8 EDIT FOLDER

```
import java.io.IOException;

import java.io.PrintStream;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;



import connect.ConnectionProvider;



public class Serv_EditFolder extends HttpServlet

{



   public Serv_EditFolder()

   {

   }



   public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

       throws ServletException, IOException

   {
    try

    {

        //Class.forName("oracle.jdbc.driver.OracleDriver");

        //con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system", "ojasvi");



        con=ConnectionProvider.getConnection();



        st = con.createStatement();

        scon = getServletContext();

        HttpSession httpsession = httpservletrequest.getSession(true);

        uname = httpsession.getValue("name").toString();

        of = httpservletrequest.getParameter("fol");

        System.out.println("of: " + of);

        ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

        servletoutputstream.println("<html><body bgcolor=white
background='INDTEXTB.JPG' text=blue>");

        servletoutputstream.println("<form name=for method=post
action='Serv_ModifyFolder'>");

        servletoutputstream.println("<h3>Changing the name of '" + of + "'</h3><br>");

        servletoutputstream.println("");
```

```
            servletoutputstream.println("<b>Folders New Name</b>  :   <input type=text name=newfol value=''><br><br>");

            servletoutputstream.println("<input type=submit name=su value='MODIFY-FOLDER'>");

            servletoutputstream.println("<input type=hidden name=hid value='" + of + "'>");

            servletoutputstream.println("</form></body></HTML>");

        }

        catch(Exception exception)

        {

            exception.printStackTrace();

        }

    }



    Statement st;

    Connection con;

    ResultSet rs;

    ServletContext scon;

    String uname;

    String of;

}
```

5.9 CREATE TABLE

```
package create;



import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.Statement;

import java.util.Scanner;



import connect.ConnectionProvider;



public class CreateTable

{

        public static void create(String path)

        {



                try

                {

                path =path+"\\"+"tables_sql.txt";
                System.out.println("******Path of
Commands***************"+path);

                FileInputStream fin=new FileInputStream(path);

                Scanner sc=new Scanner(fin);

                sc.useDelimiter("/");

                String name=" ";



                Connection con=ConnectionProvider.getConnection();



                Statement st=con.createStatement();



                int i=1;



                while(sc.hasNext())

                {

                        name=sc.next();

                        st.executeUpdate(name);

                        System.out.println(" Token "+ i++ +"  "+name);

                }

                }
                catch(Exception e)

                {

                        e.printStackTrace();

                }



        }

}
```

5.10 MODIFY FOLDER

```
import java.io.IOException;

import java.io.PrintStream;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;



import connect.ConnectionProvider;



public class Serv_ModifyFolder extends HttpServlet

{



    public Serv_ModifyFolder()

    {

    }



    public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

        throws ServletException, IOException

    {
        try

        {

            //Class.forName("oracle.jdbc.driver.OracleDriver");

            //con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system", "ojasvi");



            con=ConnectionProvider.getConnection();





            st = con.createStatement();

            scon = getServletContext();

            HttpSession httpsession = httpservletrequest.getSession(true);

            uname = httpsession.getValue("name").toString();

            ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

            nf = httpservletrequest.getParameter("newfol");

            System.out.println("new folder:" + nf);

            String s = httpservletrequest.getParameter("hid");

            System.out.println("old folder:" + s);

            int i = st.executeUpdate("update folders set folder='" + nf + "' where uname='" +
uname + "' and folder='" + s + "'");
```

```
            System.out.println("r :" + i);
            if(i > 0)
            {
                servletoutputstream.println("<html><body bgcolor=white
background='INDTEXTB.JPG' text=blue><h2><i>One record updated with latest
modifications</i></h2></body></html>");
                httpservletresponse.setHeader("Refresh", "2;URL='Serv_Folder'");
            }
        }
        catch(Exception exception)
        {
            exception.printStackTrace();
        }
    }


    Statement st;
    Connection con;
    ServletContext scon;
    String uname;
    String nf;
}
```

5.11 Compose2

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;



import connect.ConnectionProvider;



public class Serv_Compose2 extends HttpServlet

{



   public Serv_Compose2()

   {

      con = null;

      st = null;

      scon = null;

      rs = null;

      sos = null;

   }
```

```
    public void service(HttpServletRequest httpservletrequest, HttpServletResponse httpservletresponse)
        throws ServletException, IOException
    {
        try
        {
            //Class.forName("oracle.jdbc.driver.OracleDriver");
            //con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "system", "ojasvi");


            con=ConnectionProvider.getConnection();

            st = con.createStatement();
            scon = getServletContext();
            String s = null;
            String s1 = null;
            HttpSession httpsession = httpservletrequest.getSession(true);
            s = httpsession.getValue("name").toString();
            s1 = httpsession.getValue("filename").toString();
            ServletOutputStream servletoutputstream =
    httpservletresponse.getOutputStream();

            servletoutputstream.println("<html><title>Intranet Mailing System</title>");

            servletoutputstream.println("<body bgcolor=white
    background='INDTEXTB.JPG' text=blue><center><img src=adv3.gif
    border=0></center>");

            servletoutputstream.println("<pre>");

            servletoutputstream.println("<form name=form4 method=post
    action='Serv_Send'enctype='multipart/form-data'>");

            servletoutputstream.println("To            :<input type=text name=to value=
    "><br>");

            servletoutputstream.println("Subject       :<input type=text name=subject
    value="><br>");

            servletoutputstream.println("Cc            :<input type=text name=cc
    value="><br>");

            servletoutputstream.println("Bcc           :<input type=text name=bcc
    value="><br><br>");

            servletoutputstream.println("<table><tr><td><a href=Upload.html><font
    face=verdana><h5>Attach Files</h5></font></a></td><td><input type=text name=file
    value=" + s1 + "/></td></tr></table>");

            servletoutputstream.println("<textarea name=maildata rows=10
    cols=60></textarea><br>");

            servletoutputstream.println("<input type=submit name=submit1
    value='SEND'><br>");
```

```
      rs = st.executeQuery("select actname,emailid from address where uname='" + s
+ "'"");

      if(rs.next())

      {

        servletoutputstream.println("<SELECT name=se onclick=getName(value)
style=\"LEFT: 400px; POSITION: absolute; TOP: 160px; BACKGROUND-
COLOR:#b464ff;width:130px;FONT-WEIGHT:BOLD\"");

        servletoutputstream.println("SIZE=5 name=List1 value=\"List1\">");

        do

          servletoutputstream.println("<OPTION value=" + rs.getString(2) + " >" +
rs.getString(1) + "</OPTION>");

        while(rs.next());

        servletoutputstream.println("</SELECT>");

      }

      servletoutputstream.println("</form></body></html>");

      servletoutputstream.println("<script language=javascript>");

      servletoutputstream.println("function getName(Myname){");

      servletoutputstream.println("document.form4.to.value=Myname");

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

    }

    catch(Exception exception)

    {
            exception.printStackTrace();

        }

    }



    Connection con;

    Statement st;

    ServletContext scon;

    ResultSet rs;

    ServletOutputStream sos;

}
```

5.12 Serv_Enter

```
import java.io.IOException;

import java.sql.*;

import javax.servlet.*;

import javax.servlet.http.*;



import connect.ConnectionProvider;



public class Serv_Enter extends HttpServlet

{



    public Serv_Enter()

    {

        col = 1;

    }



    public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

        throws ServletException, IOException

    {
        try

        {

          //  Class.forName("oracle.jdbc.driver.OracleDriver");

        //   con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"system", "ojasvi");




            con=ConnectionProvider.getConnection();



            st = con.createStatement();

            scon = getServletContext();

            HttpSession httpsession = httpservletrequest.getSession(true);

            uname = httpsession.getValue("name").toString();

            ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

            String s = httpservletrequest.getParameter("se");

            servletoutputstream.println("<html><head><title>Welcome To Intranet Mailing
System</title></head>");

            servletoutputstream.println("<script>function d()");

            servletoutputstream.println("{document.f.action='Serv_DeleteCkd';");

            servletoutputstream.println("document.f.submit();}");
            servletoutputstream.println("function e()");

            servletoutputstream.println("{document.f.action='Serv_Enter';");

            servletoutputstream.println("document.f.submit();}");

            servletoutputstream.println("</script></head>");

            servletoutputstream.println("<body bgcolor=white
background='INDTEXTB.JPG' text=blue>");

            servletoutputstream.println("<center><blink><h1><b>" + s +
"Screen</b></h1></blink></center><br>");

            servletoutputstream.println("<h3><i>" + s + " of " + uname + " </i></h3>");

            servletoutputstream.println("<form name=f action='Serv_MoveMessage'>");

            servletoutputstream.println("<input type=button name=delete value='Delete'
onClick='d()' >");

            servletoutputstream.println("<input type=reset name=deselect
value='DeSelect'><br><br>");

            servletoutputstream.println("<h4><i>Messages in " + s + "</i></h4>");

            servletoutputstream.println("<table border=2>");

            servletoutputstream.println("<TR
bgcolor=yellow><th>Status</th><TH>X</TH><TH>From</TH><TH>Subject</Th><
TH>Date</TH></TR>");

            st1 = con.createStatement();

            rs1 = st1.executeQuery("select mailid,mailfrom,subject,maildate,mailst from
newcompose where mailto='" + uname + "' and folder='" + s + "'");

            int i = 0;
        for(; rs1.next(); servletoutputstream.println("<tr bgcolor=" + color + "><td>" +
sta + "</td><td><input type=checkbox name=b" + i + " value=" + m + "></td><td><a
href='http://localhost:8080/servlet/Serv_SeeMessage?id=" + m + "'>" + from +
"</td></a><td>" + sub + "</td><td>" + date1 + "</td></tr>"))

        {

          i++;

          m = rs1.getInt(1);

          from = rs1.getString(2);

          sub = rs1.getString(3);

          date1 = rs1.getString(4);

          sta = rs1.getInt(5);

          if(col % 2 == 0)

            color = "pink";

          else

            color = "violet";

          col++;

        }



        servletoutputstream.println("<input type=hidden name=hid  value=" + i + ">");

        servletoutputstream.println("</form></table></body></html>");

      }
```

```
    catch(Exception exception)
    {
        exception.printStackTrace();
    }
}

Statement st;
Statement st1;
Connection con;
ServletContext scon;
ResultSet rs;
ResultSet rs1;
String uname;
String sub;
String date1;
String from;
int m;
int sta;
int col;
String color;
```

5.13 Serv_Description

```
import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.*;



public class description extends HttpServlet

{



    public description()

    {

    }



    public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

        throws IOException, ServletException

    {

        PrintWriter printwriter = httpservletresponse.getWriter();

        httpservletresponse.setContentType("Text/Html");

        HttpSession httpsession = httpservletrequest.getSession(true);
        String s = httpsession.getValue("name").toString();

        printwriter.println("<html>");

        printwriter.println("<body bgcolor='#00FA9A' text=blue width='70%'
align='center'>");

        printwriter.println("<br>");

        printwriter.println("<center><h1><font color=brown> Hi " + s + "<h2><font
color=blue>");

        printwriter.println("<font color=red>GoTo <a href='Serv_Inbox'> Inbox</a>");

        printwriter.println("<h2><b><i></i></b></h2><br><br><br><br>");

        printwriter.println("<img src='Images/MOVED.GIF' border=0>");

        printwriter.println("</body></html>");

    }

}
```

5.14 Serv_List

```
import java.io.IOException;

import javax.servlet.*;

import javax.servlet.http.*;



public class Serv_List extends HttpServlet

{



    public Serv_List()

    {

        scon = null;

        newfol = null;

        f1 = null;

        uname = null;

        sub = null;

        from = null;

        date1 = null;

        folder = null;

    }
```

```
  public void service(HttpServletRequest httpservletrequest, HttpServletResponse
httpservletresponse)

    throws ServletException, IOException

  {

    try

    {

      scon = getServletContext();

      HttpSession httpsession = httpservletrequest.getSession(true);

      uname = httpsession.getValue("name").toString();

      ServletOutputStream servletoutputstream =
httpservletresponse.getOutputStream();

      servletoutputstream.println("<html><head><title>Welcome To Intranet Mailing
System</title><!--mstheme--><link rel=stylesheet href='copy1111.css'>");

      servletoutputstream.println("<meta name=Microsoft Theme content=copy-of-
industrial 1111>");

      servletoutputstream.println("</head><body leftmargin=0
background='INDTEXTB.JPG' Topmargin=0><!--table cellpadding=16 align=center--
>");



      servletoutputstream.println("<center><pre><i><b><a href=Serv_Inbox
target=in2><img src='ims%20images/INBOX.JPG' border=0 width=108
height=40></a>");
```

```
        servletoutputstream.println("<a href=Serv_Compose target=in2><img
src='ims%20images/COMPOSE.JPG' border=0 width=108
height=40></a></b></i><font color=#FFFFFF>1</font><i><b>");

        servletoutputstream.println("<a href=Serv_Option target=in2><img
src='ims%20images/Options.jpg' border=0 width=108 height=40
lowsrc=Images%20Raj/Options.jpg></a><font color=#FFFFFF>1</font>");

        servletoutputstream.println("<pre><i><b><a href=Serv_Logout
target=_parent><img src='ims%20images/Logout.jpg' border=0 width=108
height=36></a></b></i></pre><center>");


      }

      catch(Exception exception)

      {

        exception.printStackTrace();

      }

    }


    ServletContext scon;

    String newfol;

    String f1;

    String uname;

    String sub;

    String from;
    String date1;

    String folder;

}
```

CHAPTER 6

TESTING

The development of software systems involves a service of production activities where opportunities for injection of human fallibilities are enormous. Errors may begin to occur at the very inception of the process where the objectives may be erroneously or imperfectly specified, as well as later design and development stages. Because of human inability to perform and communicate with perfection, software companies are accompanied by a quality assurance activity.

Testing is to determine errors in a software code. It is crucial element of software quality assurance and represents the ultimate review of specifications, design and coding. The increasing visibility of a software as a system element and the attendant “costs” associated with a software failure are motivating forces for well planned through testing. Usually software development organizations expend between 30 to 40 percent of total project effort on testing. . Our goal is to design a series of test cases that have a high likelihood of finding errors.

To test the software, there are so many testing techniques which provide systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

If testing is conducted successfully, it will uncover errors in the software. As the secondary benefits, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appear to have been met

6.1 TESTING PRINCIPLES

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- Tests should begin with “in the small” and progress toward testing “in the large”
- Exhaustive testing is not possible.
- To be more effective an independent third party should conduct testing.

6.2. TESTING TECHNIQUES

Software design is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Once source code has been generated, software must be tested to uncover as many as errors as possible before delivery to the customer. Our goal is to design a series of test cases that have a high likelihood of finding errors. To test the software, there are so many testing techniques which provide systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in program function, behavior and performance.

If testing is conducted successfully, it will uncover errors in the software. As the secondary benefits, testing demonstrates that software functions appear to be working according to specification, that behavioral and performance requirements appear to have been met. The software can be tested by one of the two ways: -

A. Knowing the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational while at the same time searching for errors in each function.

B. Knowing the internal working of the product, tests can be conducted to ensure that internal operations are performed according to specifications and all internal components have been adequately exercised.

The first approach is called white – box testing and the second, black – boxtesting.

6.2.1. WHITE BOX TESTING

White box testing is a test case design method that uses the control structural of the procedural design to derive test cases. It is also called glass box testing. Using this method, we can derive test cases that-

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides Executes all loops at their boundaries and within their operational bounds.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

The purpose of any security testing method is to ensure the robustness of a system in the face of malicious attacks or regular software failures. White box testing is performed based on the knowledge of how the system is implemented. White box testing includes analyzing data flow, control flow, information flow, coding practices, and exception and error handling within the system, to test the intended and unintended software behavior. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities.

White box testing requires access to the source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during the unit testing phase.

6.2.2 .BLACK BOX TESTING

Black – box testing focuses on the functional requirements of the software i.e. it enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. It is also called behavioral testing.

It attempts to find errors in the following areas:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Performance errors
- Initialization errors

This software is developing as a product to be used by many customers, it is impractical to perform formal acceptance tests with each one. So out software product builders will use a process called Alpha and Beta testing to uncover errors that only the end-user seems able to find.

Advantages and Disadvantages:-

Advantages of Black Box Testing

- more effective on larger units of code than glass box testing
- tester needs no knowledge of implementation, including specific programming languages
- tester and programmer are independent of each other
- tests are done from a user's point of view

- will help to expose any ambiguities or inconsistencies in the specifications

- test cases can be designed as soon as the specifications are complete

Disadvantages of Black Box Testing

- only a small number of possible inputs can actually be tested, to test every possible input stream would take nearly forever

- without clear and concise specifications, test cases are hard to design

- there may be unnecessary repetition of test inputs if the tester is not informed of test cases the programmer has already tried

- may leave many program paths untested

- cannot be directed toward specific segments of code which may be very complex (and therefore more error prone)

- most testing related research has been directed toward glass box testing

6.2.3. ALPHA TESTING

The Alpha test is conducted at the developer’s site by a customer. The software is used in a natural setting with the developer “ looking over the shoulder” of the user and recording errors and usage problem. Alpha test are conducted in controlled environment.

6.2.4. BETA TESTING

The Beta-test is conducted at one or more customer sites by the end-user of the software.Unlike Alpha testing, the developer is generally not present. Therefore, the Beta-test is a “live” application of the software in an environment that cannot be controlled by the developer. The customer records all problems (real or imagined) that are encountered during Beta testing and reports there to the developer at regular intervals. As a result of problems reported during Beta-tests, software engineers make modifications and then prepare for release of the software product to the entire customer base.

In this project we will performed incremental testing in which components and subsystem of the system are tested separately before integrating them to form the system from system testing.

6.3. TESTING STRATEGIES

Designing effective test cases is important but so is the strategy we use to execute them. A strategy for software test case design methods that result in the successful construction of software. The strategy provides a road map that describes the steps to be conducted as a part of testing.

There are a number of testing strategies, which have the following generic characteristics:

- Testing begins at the component level and works “outward” toward the integration of the entire computer–based system.

- Different testing techniques are approximate at different points in time.

- Testing is conducted by the developer of the software and (for large projects) an independent test group.

- Testing and debugging are different activities, but debugging must be accommodated in any testing strategy.

6.4. TESTING STRATEGY

Initially, system engineering defines the role of software and leads to software requirement analysis where the information domain, function, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to the design and finally to coding. There are a number of testing strategies, which are given below: -

6.4.1. UNIT TESTING

In the unit testing interfaces, local data structures, boundary conditions, independent paths, error-handling paths are tested. Test cases should be design to uncover errors due to erroneous computations, incorrect comparisons, or improper control flow. For this purpose basis path and loop testing is done. After source level code has been developed, reviewed and verified for correspondence to component level design, unit test case design begins. In unit test application 'drivers' are developed which are programs, accept test case data, passes such data to the component to be tested and prints relevant results. 'Stubs' are also developed which serve to replace modules, that are subordinate the component to be tested.

Six Rules of Unit Testing:-

1. Write the test first
2. Never write a test that succeeds the first time
3. Start with the null case, or something that doesn't work
4. Don't be afraid of doing something trivial to make the test work
5. Loose coupling and testability go hand in hand
6. Use mock objects

6.4.2. LIMITATIONS OF UNIT TESTING

Testing, in general, cannot be expected to catch every error in the program. The same is true for unit testing. By definition, it only tests the functionality of the units themselves. Therefore, it may not catch integration errors, performance problems, or other system-wide issues. Unit testing is more effective if it is used in conjunction with other software testing activities.

Like all forms of software testing, unit tests can only show the presence of errors; it cannot show the absence of errors.

Software testing is a combinatorial problem. For example, every boolean decision statement requires at least two tests: one with an outcome of "true" and one with an outcome of "false". As a result, for every line of code written, programmers often need 3 to 5 lines of test code. Therefore, it is unrealistic to test all possible input combinations for any non-trivial piece of software without an automated characterization test generation tool such as JUnit Factory used with Java code or many of the tools listed in List of unit testing frameworks.

To obtain the intended benefits from unit testing, a rigorous sense of discipline is needed throughout the software development process. It is essential to keep careful records, not only of the tests that have been performed, but also of all changes that have been made to the source code of this or any other unit in the software. Use of a version control system is essential. If a later version of the unit fails a particular test that it had previously passed, the version-control software can provide a list of the source code changes (if any) that have been applied to the unit since that time.

It is also essential to implement a sustainable process for ensuring that test case failures are reviewed daily and addressed immediately. If such a process is not implemented and ingrained into the team's workflow, the application will evolve out of sync with the unit test suite—- increasing false positives and reducing the effectiveness of the test suite.

6.4.3. INTEGRATION TESTING

Integration testing is systematic technique for constructing the program structure while at the same time conducting the tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. There are two types of integration – Bottom up integration and Top down integration. Regression and smoke testing are done in integration testing strategy.

6.4.4. VALIDATION TESTING

Next step is the validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. At the culmination of integration testing, software is completely assembled as a package, interfacing errors has been uncovered and corrected, and a final series of software tests i.e. validation testing begins. Validation succeeds when software functions in a manner that can be reasonably expected by the customer.

Software validations can be achieved through a series of black-box tests that demonstrate conformity with requirements. Configuration review, alpha testing and beta testing techniques are conducted to enable the customer to validate all requirements. Acceptance tests are conducted by the end-users over a period of weeks or months.

6.4.5. SYSTEM TESTING

Finally, we arrive at system testing where the software and other system elements are tested as a whole. System testing verifies that all elements mesh properly and that overall system function / performance is achieved. Ultimately software is incorporated with other system elements and a series of system integration and validation tests are conducted.

6 .5. TEST CASES

TEST CASE-1

TEST NO.	:	1
TEST TYPE	:	UNIT TESTING
INPUT	:	PASSWORD
BJECTIVE	:	CHECKING PASSWORD SECURITY
EXPECTED OUTPUT	:	ACCESS TO AUTHORIZED USERS ONLY
ACTUAL OUTPUT	:	PASSWORD SECURITY SUCCESSFUL
RESULT	:	ACCESS TO ONLY AUTHORIZED USERS

TEST CASE-2

TEST NO.	:	2
TEST TYPE	:	UNIT TESTING
INPUT	:	DATA ENTRY TO REGISTER FORM
OBJECTIVE	:	TO STORE VALID DATA IN THE TABLE
EXPECTED OUTPUT	:	SUCCESSFUL DATA STORAGE
ACTUAL OUTPUT	:	SUCCESSFUL DATA STORAGE
RESULT	:	DATA STORED

TEST CASE : 3

TEST NO. : 3

TEST TYPE : UNIT TESTING

INPUT : DATA ENTRY TO VARIFICATION OF USER TYPE

OBJECTIVE : TO OPEN USER OR ADMIN PAGE

EXPECTED OUTPUT : SUCCESSFUL OPEN USER HOME PAGE

ACTUAL OUTPUT : SUCCESSFUL OPEN USER HOME PAGE

RESULT : ACKNOWLEGMENT

TEST CASE-4

TEST NO. : 4

TEST TYPE : UNIT TESTING

INPUT : DATA ENTRY MAIL DETAILS

OBJECTIVE : FOR MAIL

EXPECTED OUTPUT : SUCCESSFUL MAIL INSERT IN DATABASE

ACTUAL OUTPUT : SUCCESSFUL MAIL INSERT IN DATABASE

RESULT : ACKNOWLEGMENT

TEST CASE -5

TEST NO.	: 5
TEST TYPE	: UNIT TESTING
INPUT	: MAIL CATEGORY
OBJECTIVE	: TO SAVE DATA INTO THE DATABASE
EXPECTED OUTPUT	: SAVE DETAILS
ACTUAL OUTPUT	: SAVE DETAILS
RESULT	: OK

TEST CASE – 6

TEST NO	: 6
TEST TYPE	: UNIT TESTING
INPUT	: SELECT MESSAGE
OBJECTIVE	: FOR CHAT
EXPECTED OUTPUT	: BETTER MATCHING PROJECT
ACTUAL OUTPUT	: BETTER MATCHING PROJECT
RESULT	: OK

6 .6. DEBUGGING AND CODE IMPROVEMENT

Since the modular approach is used to design the software. This approach uses the object oriented design methods. Hence the code is improved due to the above reason. Because of object oriented programming, the features like modularity and reusability can be achieved in the software.

In my project I have used various controls like command buttons, text box and similar other elements from various class.

The process of testing gives symptoms, and a program’s failure is a clear symptom of the presence of the error. After getting a symptom, we begin an investigation to localize the error, that is to find out which module or interface is causing it. Then that section of the code is to be studied to determine the cause of the problem. This process is called ‘Debugging’. Hence, debugging is the activity of locating and correcting errors.

The following errors has been debugged during the creation of the project-

S.No.	Bug Description	Cause of Bug	Time required to remove it
1.	Update error	Database Connectivity problem	10 Minutes
2	Retrieve error	Sql error	20 minute
3.	Database Insert Error	Data Type not match	20 Minutes
4.	Record Not Found	DB error	10 Minutes

CHAPTER 7

MAINTENANCE

7.1. INTRODUCTION

Software maintenance refers to the process of modifying and updating a software system after it has been delivered to the customer. This can include fixing bugs, adding new features, improving performance, or updating the software to work with new hardware or software systems.

The goal of software maintenance is to keep the software system working correctly, efficiently, and securely, and to ensure that it continues to meet the needs of the users.

Maintenance is required to maintain a building's initial performance capacity. Without maintenance, performance will not meet the demand and eventually will drop below the limit of acceptance of residents. In practice, both the demand and the limit of acceptance will gradually rise over time as a result of improved technology, rising standards, and growing prosperity.

Improvement and renewal are required to answer the accordingly rising expectations. As a result, the total life cycle costs will generally be a multiple of the initial building costs. Maintenance is a combination of all technical and associated administrative actions during the service life to retain a building or its parts in a state in which it can perform its required functions.

7.2. TYPES OF WEBSITE MAINTENACE SERVICES

7.2.1. UPDATING WEBSITE SOFTWARE

If you are using any kind of content management system like Word Press or any kind of script like PHP, it needs to always be up to date. Word Press usually updates itself automatically but there are instances where it might need to be updated manually.

Additionally, if you are using a website host, a web service maintenance provider will ensure that they are updating their core server software on a regular basis. For instance, they will check if things like FTP service running on the server are up to date. If however, you run your own web server, they will manually test and apply updates.

7.2.2. IMPROVING WEBSITE SPEED

Did you know that twenty five percent of users navigate out of a website if it takes more than three seconds to load? Page loading is obviously an important part of the user experience. However, we tend to let it slide in order to accommodate new nifty functionality or aesthetic website design.

In reality, web visitors do not care about all the bells and whistles. They care about accessing the information they want as fast as possible. Another important factor to consider is how page load speed affects your search engine rankings. A slow website increases bounce rate and if people are navigating out of your web page as fast as they got there, then Google will think your website isn't important so they will push it further down the rankings.

7.2.2.1. FIXING HTML ERRORS

HTML Code isn't easy especially if you know nothing about website coding. You need to pay attention to detail to ensure that it's working well. What might seem like a minor error could lead to a number of problems such as funky looking pages, missing multimedia, etc. Reliable website support should be able to inspect, find and fix issues related to HTML.

7.2.2.2. BACKING UP FILES

Imagine how frustrating it would be if you were to lose all your website's essential files.

This includes images, pages, blog posts, plugging, etc. If you run your website on Word Press, you might have a few automated backup options. Website maintenance services ensure that file and database backups is automatically performed at least on a weekly basis.

7.2.2.3. DEVELOPING NEW CONTENT

Your content strategy might be on point but it's always a good idea to improve it by creating new types of content. Your web content also needs to be up to date. If you wrote a post that was published years ago, it might not be relevant right now and some of the information might need to be updated.

7.2.2.4.SEARCH ENGINE OPTIMIZATION

Regularly checking your website for areas where search ability can be improved can go a long way to improving search engine optimization. Your blog posts can be altered to include popular and competitive keywords. Your landing pages can also be updated to include more relevant and timely keywords. Even you're "About Us" page can be updated regularly to take advantage of popular keywords in your industry. Keyword research is also important when creating new content for your website.

As part of the SEO strategy, it's also important to identify structural issues within your website that may affect how search engines view your site.

7.2.2.5. ENSURING DESIGN CONSISTENCY ACROSS ALL PAGES

Your website should function properly in all the latest versions of major website browsers like Chrome, Safari, Firefox, etc. These browsers update frequently and if your website doesn't adapt to the updates it might not show up or it might not look the way you want it to. This could negatively affect your business and you're branding. It's also important to ensure that you have a responsive website design that looks good on mobile devices.

Another factor to consider about your website's design is uniformity. If you have a lot of branding elements that represent your company, they should be included uniformly across all your pages.

7.3. FIXING BROKEN LINES

Every link on your website should lead exactly to the right place but sometimes links can go bad. A site or resource you linked to earlier might disappear, a page on your website might become unavailable or you might move a post and forget to update others that link to it.

Broken links can be detrimental to your website. They could give visitors the impression that:

- Your website is not user friendly
- You don't regularly update your website
- You are not credible

7.4. REVIEWING YOUR WEBSITE ANALYTICS

Your website's performance needs to be monitored to ensure that everything is working as it should. An experienced website support specialist will examine high level metrics like how many new and returning visitors your website has received for the past week or which blog posts are the most read. They will then make adjustments your website accordingly to ensure all the metrics are at their peak.

This is just a minimum starting point of what website maintenance is all about. Some of these tasks can be overly time consuming. Additionally, if you do not know what you are doing, you could end up doing more damage. The better option is to hire professional website maintenance services.

CHAPTER 8

LITERATURE REVIEW

The Internet Mailing System is a crucial component of modern communication, enabling individuals and organizations to exchange messages and information quickly and conveniently. This literature review aims to provide a comprehensive overview of the evolution, functionalities, benefits, challenges, and future prospects of the Internet Mailing System.

Evolution of the Internet Mailing System:

The Internet Mailing System has evolved significantly since its inception. Initially, simple protocols like POP (Post Office Protocol) and SMTP (Simple Mail Transfer Protocol) facilitated basic email functionality. Over time, advancements such as MIME (Multipurpose Internet Mail Extensions) improved email capabilities, allowing the transmission of attachments, multimedia content, and HTML formatting.

Functionalities of the Internet Mailing System:

a) Message Composition and Formatting: Users can create and format emails using various tools, including rich text editors, HTML coding, and inline attachments.

b) Message Transmission: SMTP is primarily responsible for routing and transmitting messages across the Internet.

c) Message Retrieval: POP, IMAP (Internet Message Access Protocol), and web-based interfaces enable users to retrieve and manage their emails from remote servers.

d) Filtering and Sorting: Email clients and servers often employ filters and rules to categorize, sort, and prioritize incoming messages.

e) Security Measures: Encryption, digital signatures, spam filters, and antivirus scanners ensure the integrity, confidentiality, and authenticity of email communications.

Benefits of the Internet Mailing System:

a) Speed and Efficiency: Emails enable near-instantaneous delivery, allowing for quick and efficient communication across vast distances.

b) Cost-Effectiveness: Email eliminates the need for physical mail, reducing postage and paper costs.

c) Global Reach: The Internet Mailing System connects individuals and organizations worldwide, facilitating international communication effortlessly.

d) Documentation and Archiving: Emails serve as a written record of conversations, aiding in documentation and future reference.

e) Flexibility and Accessibility: Users can access their emails from various devices and locations, ensuring constant connectivity.

Challenges and Limitations:

a) Security Concerns: Email is susceptible to hacking, phishing attacks, and spam, necessitating robust security measures.

b) Email Overload: The sheer volume of emails received can lead to information overload and reduced productivity.

c) Privacy Concerns: The storage and processing of emails raise concerns regarding data privacy and surveillance.

d) Standardization and Interoperability: Different email clients and protocols can create compatibility issues, affecting seamless communication.

Future Prospects:

a) Integration with Collaborative Tools: Email systems are likely to integrate with collaborative platforms, allowing real-time document editing, task management, and video conferencing within email interfaces.

b) Enhanced Security Measures: Advancements in encryption, artificial intelligence, and machine learning will improve email security, thwarting emerging threats effectively.

c) Intelligent Filtering and Sorting: AI-based algorithms will play a crucial role in intelligent email filtering, ensuring that users receive relevant messages and reducing email overload.

d) Mobile Optimization: The increasing use of smartphones necessitates the optimization of email clients and interfaces for mobile devices, enhancing user experience and accessibility.

CHAPTER 9

CONCLUSION

In conclusion, the internet mailing system has played a pivotal role in revolutionizing communication and has become an essential tool for individuals, businesses, and organizations worldwide. The project's future scope presents numerous opportunities for further advancements and improvements in the system.

Enhanced security features will ensure the protection of user data and privacy, instilling trust in the system. Integration with artificial intelligence will bring intelligence and personalization to the system, making email management more efficient and tailored to individual needs. Integration with voice and video calling will provide a seamless communication experience within the mailing system, eliminating the need for separate applications.

The incorporation of collaborative and productivity tools will enable users to work together more effectively, improving productivity and streamlining workflows. Integration with IoT devices will extend the accessibility of the system, allowing users to manage their emails across a range of interconnected devices. An enhanced mobile experience will cater to the growing number of mobile users, ensuring a user-friendly interface and convenient access to emails on the go.

Improved search and organization capabilities will simplify email management, helping users quickly find and categorize their emails. Integration with social media platforms will consolidate communication channels, enabling users to manage both professional and social interactions within the same system. Localization and multilingual support will facilitate cross-cultural communication, making the system more inclusive and adaptable to diverse user preferences.

Furthermore, the integration of analytics and insights will empower users with valuable information about their email usage patterns, productivity metrics, and communication habits. These insights will enable users to make data-driven decisions, optimize their email management strategies, and improve overall efficiency.

In conclusion, the future scope of the internet mailing system project holds great potential for enhancing security, leveraging AI technologies, integrating with various communication modes, improving collaboration and productivity, extending device compatibility, providing an enhanced mobile experience, streamlining search and organization, integrating with social media, offering multilingual support, and delivering valuable analytics and insights. By embracing these advancements, the internet mailing system will continue to evolve, meeting the ever-changing communication needs of users and driving efficiency in personal and professional interactions.

CHAPTER 10

FUTURE SCOPE

The future scope of the internet mailing system project includes enhancing Security features, integrating artificial intelligence technologies, incorporating voice and video calling capabilities, providing collaborative and productivity tools, integrating with IoT devices, improving the mobile experience, enhancing search and organization apabilities, integrating with social media platforms, offering localization and multilingual support, and providing analytics and insights to users. These developments will shape the future of internet mailing systems, ensuring secure communication, personalized experiences, efficient collaboration, seamless device integration, and enhanced user productivity.a

REFRENCES

1 Colby, J. (2003). The Intranet As Communications Platform. In: Practical Intranet Development. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5354-9_10

2 Segal, R.B., Kephart, J.O.: MailCat: An Intelligent Assistant for Organizing E-Mail. In: Proceedings of the 3rd International Conference on Autonomous Agents, Autonomous Agents 1999, Seattle, WA, USA (1999)

3 Takkinen, J., Shahmehri, N.: CAFÉ: A Conceptual Model for Managing Information in Electronic Mail. Copyright 1998 IEEE. Published in the Proceedings of the Hawaii International Conference on System Sciences, HICSS 1998, Kona, Hawaii, January 6-9 (1998)

4 Begbie, R., Chudry, F. The Intranet Chaos Matrix: A conceptual framework for designing an effective knowledge management intranet. J Database Mark Cust Strategy Manag 9, 325–338 (2002). https://doi.org/10.1057/palgrave.jdm.3240081

5 Weik, M.H. (2000). intranet. In: Computer Science and Communications Dictionary. Springer, Boston, MA. https://doi.org/10.1007/1-4020-0613-6_9536

6 Gennai, F., Martusciello, L., Buzzi, M.: A certified email system for the public administration in Italy. In: IADIS International Conference WWW/Internet, vol. 2, pp. 143–147 (2005)

7 Gao, Y., Song, J., Gao, J., Wang, J. (2022). The Research of Spam-Mail Governance Based on the Big Data Analysis of Enterprise E-mail System. In: Li, X. (eds) Advances in Intelligent Automation and Soft Computing. IASC 2021. Lecture Notes on Data Engineering and Communications Technologies, vol 80. Springer, Cham. https://doi.org/10.1007/978-3-030-81007-8_137

8 von Behren, J.R., Czerwinski, S., Joseph, A.D., Brewer, E.A., Kubiatowicz, J.: NinjaMail: the Design of a High-Performance Clustered, Distributed E-mail System. In: Proceeding of International Workshops on Parallel Processing 2000, Toronto, Canada, August 21–24, pp. 151–158 (2000)

9 Christenson, N., Bosserman, T., Beckemeyer, D.: EarthLink Network, Inc.: A Highly Scalable Electronic Mail Service Using Open Systems. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, California (December 1997)

10 Saito, Y., Bershad, B.N., Levy, H.M.: Manageability, availability and performance in Porcupine: a highly scalable, cluster-based mail service. In: 17th ACM symposium on Operating System Review, December 1999, vol. 34(5), pp. 1–15 (1999)

11 Bertolotti, L., Calzarossa, M.C.: WORKLOAD CHARACTERIZATION OF MAIL SERVERS. In: The proceedings of SPECT 2000, Vancouver, Canada, July 16–20 (2000)

12 G. Ateniese, B. d. Medeiros and M. T. Goodrich, TRICERT: A Distributed Certified E-Mail Scheme, In ISOC 2001 Network and Distributed System Security Symposium (NDSS'01), San Diego, CA, USA, Feb 2001.

BIBLIOGRAPHY

1. Java-2 Complete Reference – by patrick Naughtan

2. Java Servlet Programming- by O'Reilly

3. Pure JavaScript -by Jason Gillam, Chariton try

4. HTML Complete -BPB publication

5 http://www.sun.com
6 http://www.coreservlets.com
7 http://www.servletside.com
8 http://w3school.com
9 http://www.google.com
10 http://www.webopedia.com
11 http://www.ddj.com