BUNDELKHAND INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of Computer Science & Engineering

Session 2016-2017

A
SUMMER TRAINING REPORT
ON

" BLOOMING-BUDS (DAY-BOARDING)"

COMPLETED

AT

"PRECURSOR INFO SOLUTION"

.::SUBMITTED BY::.

ANKRITI R. SACHAN

ROLL NO. : 1304310011

B.TECH VII SEMESTER, Computer Science & Engineering

.::SUBMITTED TO::.

Department of Computer Science & Engineering

BUNDELKHAND INSTITUTE OF ENGINEERING AND TECHNOLOGY,JHANSI

Certificate

This is to certify that **Ankriti R. Sachan** , a student of B.Tech of batch 2013-2017 has completed the Project titled **"Blooming Buds(Day-Boarding)"** being submitted for the partial fulfillment of degree of B.Tech (Computer Science & Technology).

This is to further certify that the student has attended the **Summer Training** at **PRECURSOR INFO SOLUTION, LUCKNOW(U.P.)** for the 6 weeks after the sixth semester theory exams. Her work has been satisfactory and commendable.

Dr. Yashpal Singh

Head of Department

Computer Science & Engineering

Acknowledgement

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their wholehearted co-operation and have helped me in completing this project successfully.

First of all, I would like to thank **Mrs. Swasti Agrawal, PRECURSOR INFO SOLUTIONS (LUCKNOW)** for creating opportunities to undertake me in the esteemed organization.

Special thanks to **Dr. Yashpal Singh**, Head Of Department for all the help and guidance extended to me by him in every stage. I am also grateful to all faculty members for their indomitable contribution and guidance without which the completion of this seminar report would have been impossible.

In all I found a congenial work environment in **PRECURSOR INFO SOLUTIONS LUCKNOW** and this completion of the project will mark a new beginning for me in the coming days.

ANKRITI R. SACHAN
C.S.E FINAL YEAR
ROLL NO. -1304310011

TABLE OF CONTENT

CONTENT **PAGE NO**

1.Introduction

1.1-Blooming Buds:

Day-Boarding center is the place where the care of a child is taken during the day by a person other than the child's legal guardians, typically performed by someone outside the child's immediate family. Blooming Buds is typically an ongoing service during specific periods, such as the parents' time at work. The vast majority of childcare is still performed by the parents, in-house nannies or through informal arrangements with relatives, neighbors or friends. Child care in the child's own home is traditionally provided by a nanny or by extended family members including grandparents, aunts and uncles.

The Blooming Buds industry is a continuum from personal parental care to large, regulated institutions. Some child minders care for children from several families at the same time, either in their own home (commonly known as "family Blooming Buds" in Australia) or in a specialized child care facility. Some employers provide nursery provisions for their employees at or near the place of employment. In Canada, the workforce is predominantly female (95%) and low paid, averaging only 60% of average workforce wage.

1.2 DEFINITION, ACRONYMS & ABBREVIATION

1. **HTML (Hyper Text Mark-up Language):** It is used to create static web pages.
2. **SQL (Structured Query Language):** It is a query language that allows access to data residing in a database management system.
3. **Java:** It is a platform independent programming language.
4. **HTTP (Hyper Text Transfer Protocol):** It is a transaction oriented client/server protocol between a web browser and a web server.

1.3 TECHNOLOGY USED

Server Side : Java(JDK1.8.0), Servlet2.4, JSP2.0, EL, JSTL
Client Side : HTML, JavaScript, CSS
Database : Oracle 10g
IDE : NetBeans IDE 8.0.2

1.4 REQUIREMENT SPECIFICATION

Software Specification

Developer Side

Operating System:	Windows 8.1
Application System:	NetBeans
Data Base:	Oracle !0g
Programming Language:	Java

Client Side

Operating System:	Any
Web Browser:	Any

Server Side

Operating System	Any
Web Server:	Apache Tomcat

Hardware Specification

Developer Side

Processor	Intel p4or equivalent
RAM	2GB

Client Side

Processor	Intel p4or equivalent
RAM	2GB

Server Side

Processor	Intel p4or equivalent
RAM	2GB

2. OVERALL DESCRIPTION

2.1 Project Perspective

Blooming Buds is a child care site that provide the facility of child. Every parents can register and each child has own caretaker . Caretaker also create new account and provide all details to admin ,admin verify the profile of care taker via email or message and after verification admin provide user id and password to admin . This site provide the facilities only for day.

2.2 OBJECTIVES

Blooming Buds(Day-Boarding) support the rights of the child and are committed to practice which protects children from harm.

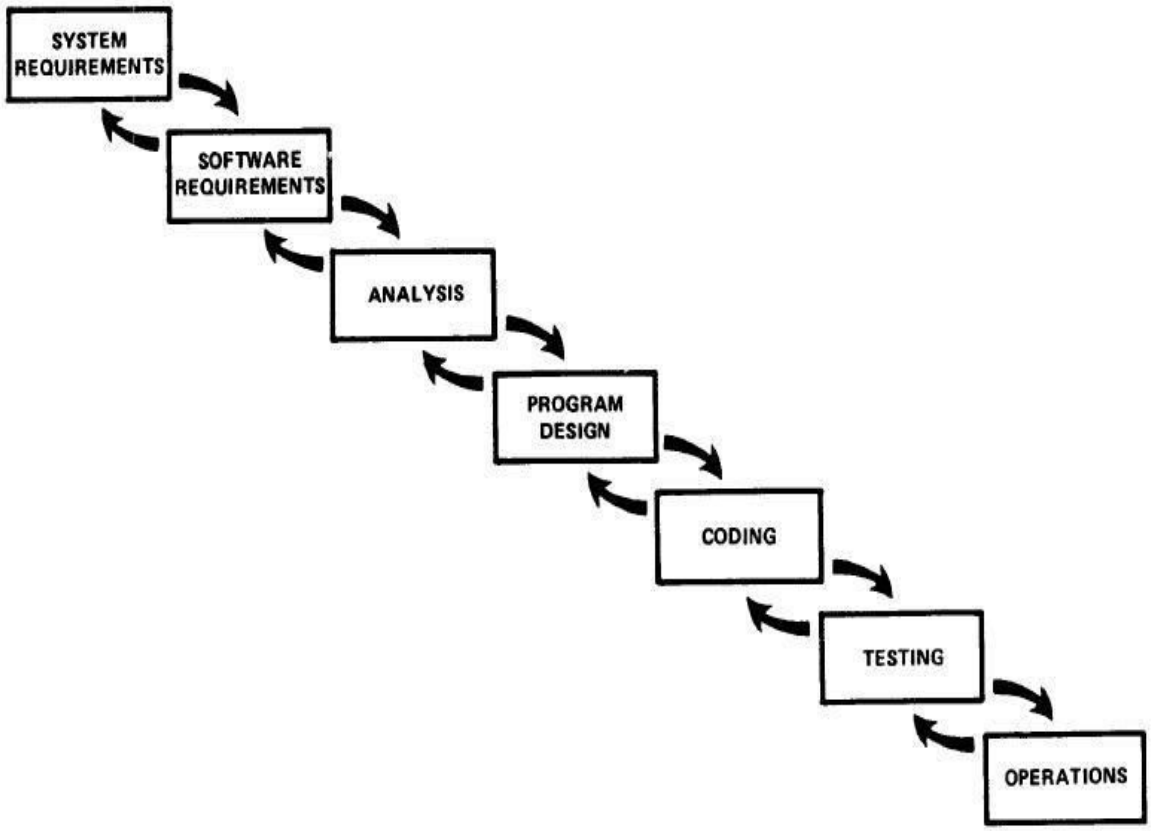
- To ensure all person providing direct care to child are vetted by a police check
- To ensure Staff caring for children are experienced, qualified, caring ,and professional.
- To provide effective management for staff, trainees and volunteers through supervision, support and training.
- To provide quality care and comfort in a safe friendly environment.
- To protect and encourage children and focus on positive behavior.
- To provide opportunity for creative and imaginative play where children explore their own identity and help build their self-esteem.
- To offer early learning program through the intervention of planned play activities that allow children to grow and enable them to reach their full potential.

2.3 Users of the system

In this Blooming buds, the users are:

1. Visitor
2. Kids
3. Administrator
4. Parents
5. Caretaker

2.4 Model used:



Description of iterative waterfall model:

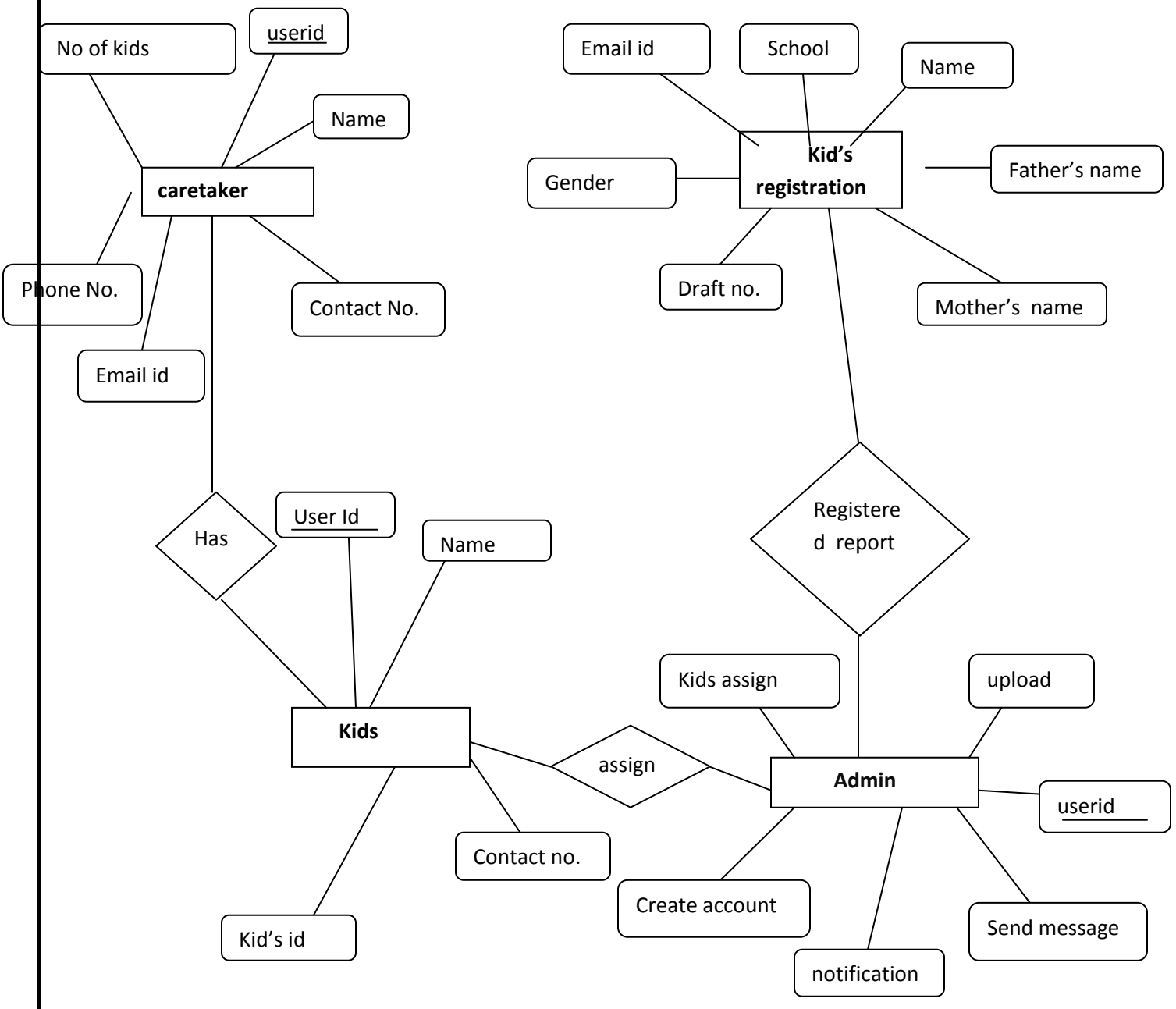
We have used Iterative waterfall model to develop this project. Iterative waterfall model uses the same concept as the Waterfall model except that it also provides an iterative approach to do any kind of modification needed even if the phase of software development life cycle is complete. In the first phase the all the requirement for the software is gathered. In the second phase the analysis of the project is sixth phase done. In the third phase work is done on the design. In the fourth phase coding is done. In the fifth phase testing is done and in the last phase maintenance is performed.

In Iterative model no stage is final stage and one can move in backward direction hence the name Iterative Waterfall. In this model we can iterate the whole SDLC lifecycle as many times as we feel like, as there is no boundation.

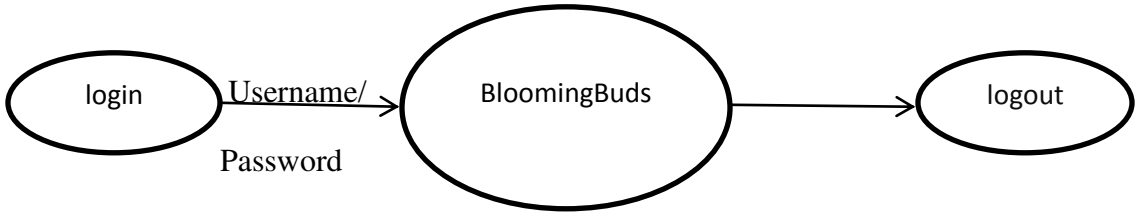
2.5 Modules Used in project:

- Sign Up – Parents can register their kids by filling a form online and paying fixed amount via draft.
- Verifying Draft after Kid's registration – Draft number is provided by the parents during kid's registration which his then verified by the admin further assignment of user id and password.
- Assignment of user id and password to parents and caretakers – Admin does this job by creating account for both kids and caretakers and conveying their user id and password through some other means like messaging, calling or email.
- Login – Parents and caretakers can login after user id and password is provided to them by admin.
- Assignment of not more than five kids to caretakers – Each caretaker cannot have more than five kids.
- Messaging module containing compose mail, inbox and sent item box – Parents can do messaging with caretaker/admin and vice versa.
- Uploading and downloading of various poems, stories etc. – Admin can upload various kinds of stories and poems and Parents can download it.
- Editing of caretaker's and kid's details – Parents and caretakers can update their details like address, email, etc.
- Query form – Anyone interested in the organization can ask any question regarding it by filling a query form in contact us page and responsible authority will contact them by either call or mail provided.
- Updating Notification on home page – Home page will show a sliding notification panel where all the latest notifications like – new updates, updating mobile numbers, etc. will be shown on top of all other notification.

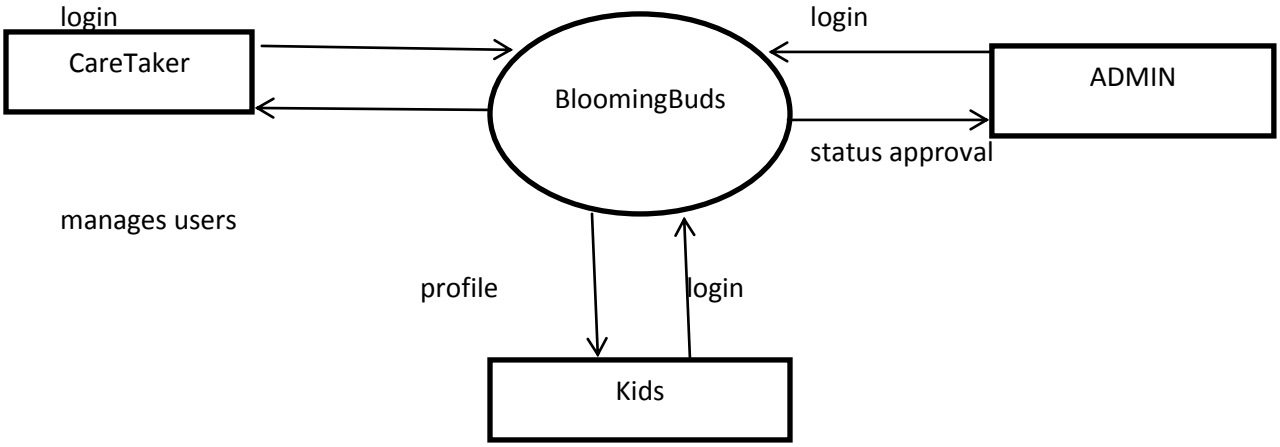
2.6 E-R Diagram



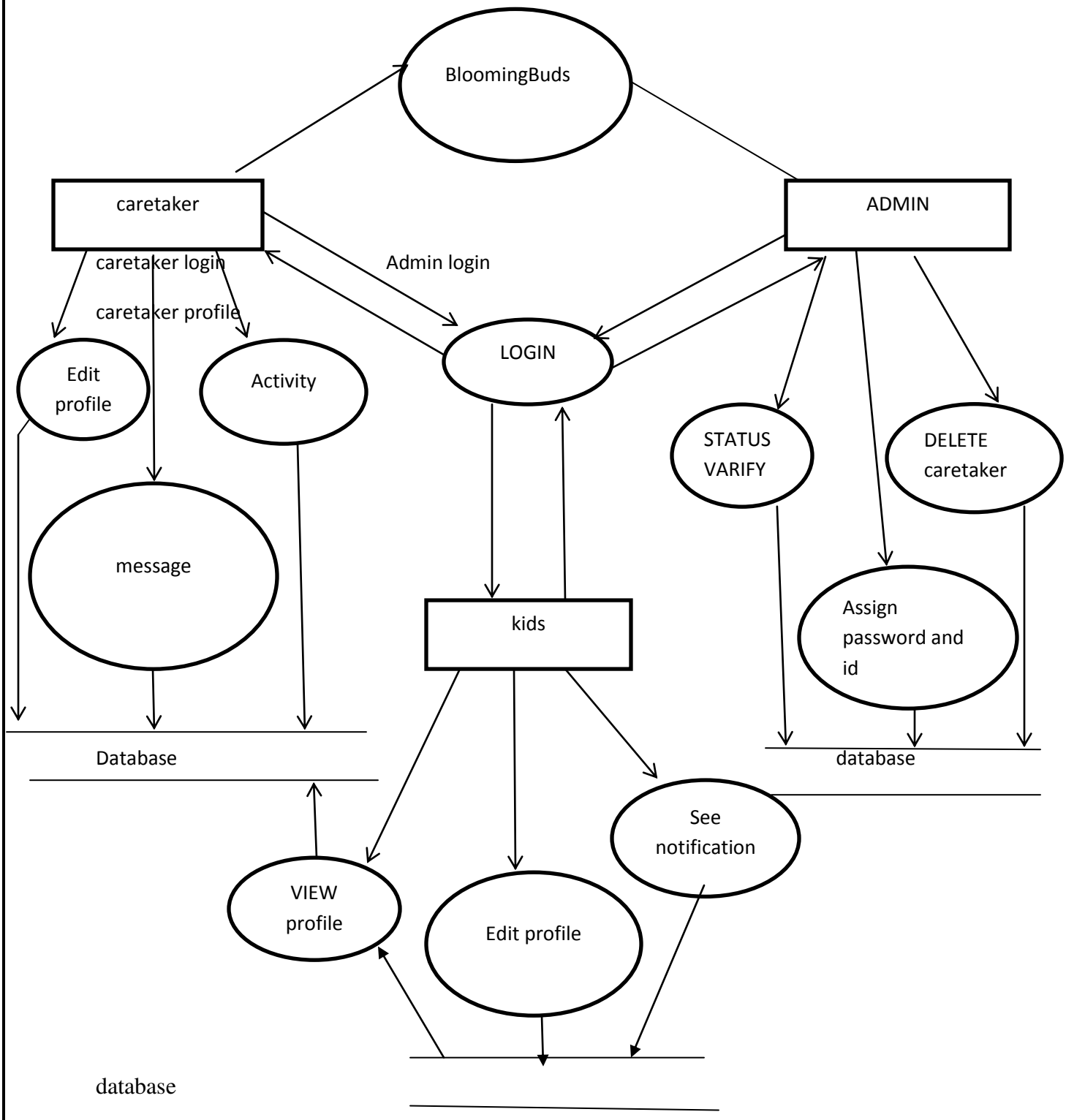
2.7 Data flow diagram



Level0 Data Flow Diagram



Level1 Data Flow Diagram



Level2 Data Flow Diagram

3. CODING

3.1 Home

```
<head>
<title>HOME</title>
<link rel="stylesheet" type="text/css" href="/bloomingBuds/css/styleA.css">
<script type="text/javascript">
var arr= new Array(3);
arr[0]="/bloomingBuds/image/logo.png";
arr[1]="/bloomingBuds/image/bbuds.jpg";
arr[2]="/bloomingBuds/image/roseday.jpg";
var idx=0;
function changeImage(){
	document.getElementById("movImg").src=arr[idx];
	idx++;
	if(idx>2)
		idx=0;
	setTimeout("changeImage()", 1500);
}
</script>
</head>
<body style="background-image: url('/bloomingBuds/image/background.jpg'); background-
repeat:no-repeat "; onload="changeImage()">
   <div id="outer">
   <div id="header">
   <%@ include file="/html/header.html" %>
    </div>
    <div id="menu">
    <ul>
       <li><a href="/bloomingBuds/jsp/home.jsp">HOME</a></li>
       <li><a href="/bloomingBuds/html/aboutus.html">ABOUT US</a></li>
       <li><a href="/bloomingBuds/html/gallery.html">GALLERY</a></li>
       <li><a href="/bloomingBuds/jsp/registration.jsp">SIGN UP</a></li>
       <li><a href="/bloomingBuds/jsp/login.jsp">SIGN IN</a></li>
       <li><a href="/bloomingBuds/jsp/contactus.jsp">CONTACT US</a></li>
    </ul>
    </div>
    <div id="body">
    <div id ="img">
    <img src="/bloomingBuds/image/bb.jpg" id="movImg">
    </div>
    <div style="height: 300px; width: 1000px">
    <div id="vid">
    <video style="width: 550px; height: 300px" src="/bloomingBuds/video/video.mp4"
loop="loop" autoplay="autoplay" controls="controls"> </video></div>
    <div id="notif">
    <h3 style="font-family: cursive;">Notifications</h3>
    <marquee direction="up" height="120px" width="950px">
    <%Connection con = Dboperation.createConnection();
    String strnf="select notification from notif order by notifid desc limit 10";
    PreparedStatement ps = con.prepareStatement(strnf);
```

```
    ResultSet rs = ps.executeQuery();
    while(rs.next())
    {%>
    <ul>
    <li><%=rs.getString("notification") %></li>
    </ul>
    <%}%>
    </marquee>
    </div>
    </div>
    <div id="footer">
    <%@include file= "/html/footer.html"%>
    </div>
  </body>
</html>
```

3.2 Registration

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
            String name = request.getParameter("kname");
            String add = request.getParameter("kadd");
            String email = request.getParameter("kmail");
            long phone = Long.parseLong(request.getParameter("kphone"));
            String gender = request.getParameter("kgen");
            String mother = request.getParameter("mother");
            String father = request.getParameter("father");
            String school = request.getParameter("school");
            String draft = request.getParameter("draft");
            String date = request.getParameter("date");

            String strreg="insert into
kidregistration(name,address,email,phoneno,gender,mothername,fathername,schoolname,dr
aftno,date,status,assignstatus,caretakerid) values(?,?,?,?,?,?,?,?,?,?,?,?,?)";

            con = Dboperation.createConnection();

            try{
            ps = con.prepareStatement(strreg);
            ps.setString(1, name);
            ps.setString(2, add);
            ps.setString(3, email);
            ps.setLong(4, phone);
            ps.setString(5, gender);
            ps.setString(6, mother);
            ps.setString(7, father);
            ps.setString(8, school);
            ps.setString(9, draft);
            SimpleDateFormat sd = new SimpleDateFormat("yyyy-mm-dd");
            try {
                  java.util.Date d = sd.parse(date);
                  long ld = d.getTime();
                  java.sql.Date s = new java.sql.Date(ld);
                  ps.setDate(10, s);
            } catch (ParseException e) {
```

```
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                }


                ps.setString(11, "false");
                ps.setString(12, "not assigned");
                ps.setString(13, "none");

                System.out.println(ps);
                int rowReg = ps.executeUpdate();

                if(rowReg>0){
                        request.setAttribute("info","You are registered successfully.
Userid and password will be provided on your email after verification.");
                        RequestDispatcher
rd=request.getRequestDispatcher("/jsp/regmessage.jsp");
                        rd.forward(request, response);
                }
                }
                catch(SQLException se){
                        System.out.println(se);
                }
                finally {
                        try {
                                ps.close();
                        } catch (SQLException e) {
                                // TODO Auto-generated catch block
                                e.printStackTrace();
                        }
                }
        }
```

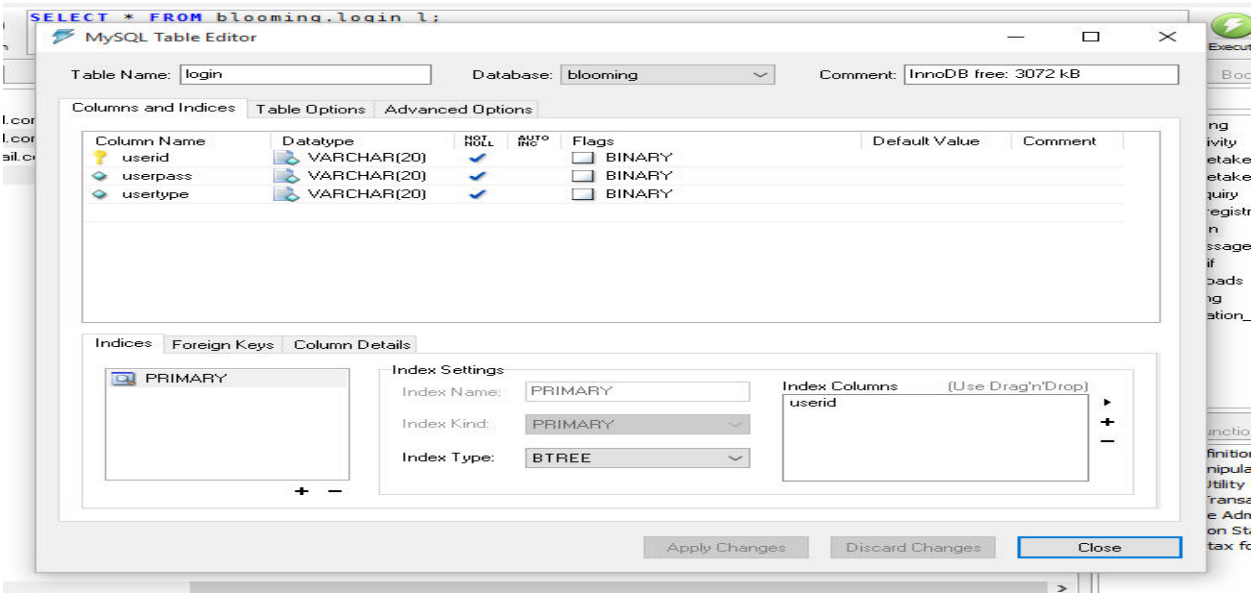
3.3 Login

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
                String uid=request.getParameter("txtid");
                String upass=request.getParameter("txtpass");
                String strsql="select * from login where userid=? and userpass=?";
                con=Dboperation.createConnection();
                try{
                        ps=con.prepareStatement(strsql);
                        ps.setString(1, uid);
                        ps.setString(2, upass);
                        rs=ps.executeQuery();
                        if(rs.next()){
                                HttpSession hs=request.getSession();
                                hs.setAttribute("log", uid);
                                String ut=rs.getString("usertype");
                                if(ut.equals("admin")){

        response.sendRedirect("/bloomingBuds/jsp/admin.jsp");
                                }
                                if(ut.equals("user")){

        response.sendRedirect("/bloomingBuds/jsp/parent.jsp");
                                }
```

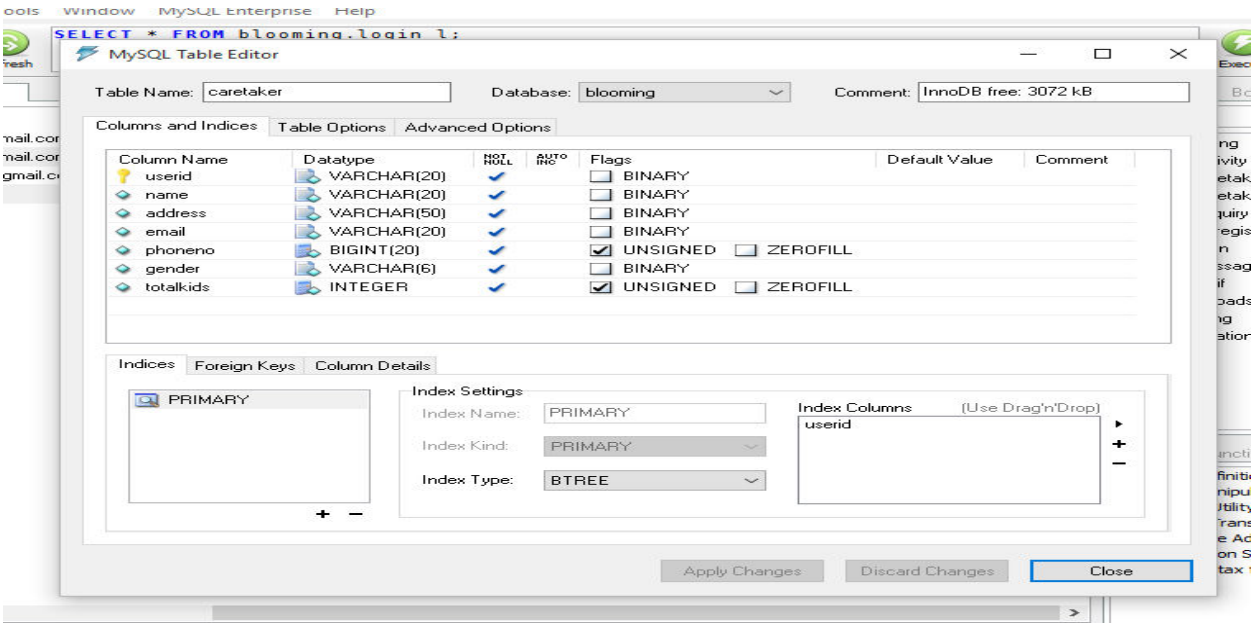
```
                        if(ut.equals("caretaker")){
                                String strsql1="select userid from caretaker where
userid=?";
                                con=Dboperation.createConnection();
                                try{
                                        ps=con.prepareStatement(strsql1);
                                        ps.setString(1, uid);
                                        rs=ps.executeQuery();
                                        if(rs.next()){

        response.sendRedirect("/bloomingBuds/jsp/ctprofile.jsp");
                                        }
                                        else{

        response.sendRedirect("/bloomingBuds/jsp/createprofile.jsp");
                                        }
                                }
                                catch(SQLException se){
                                        System.out.println(se);
                                }
                        }

                }

                else{
                        request.setAttribute("msg", "invalid userid and
password");
                        RequestDispatcher
rd=request.getRequestDispatcher("/jsp/login.jsp");
                        rd.forward(request, response);
                }
```

4. TABLE USED:

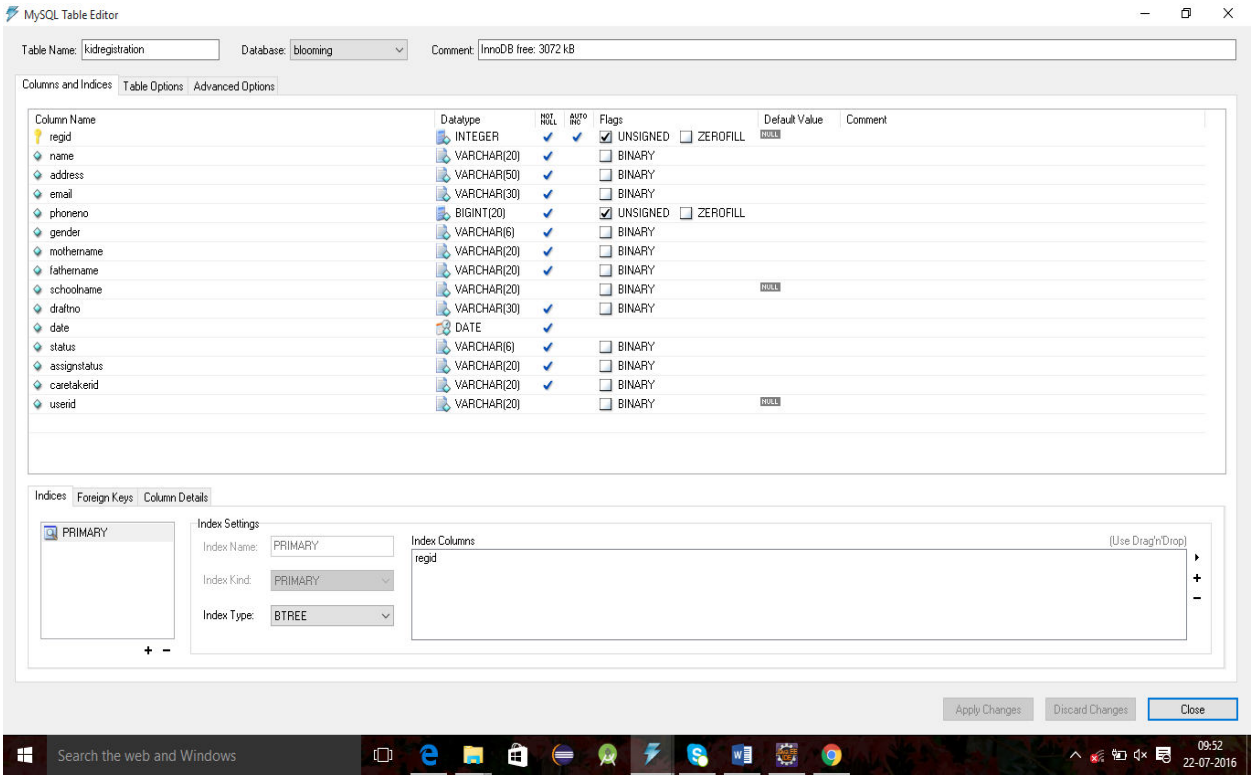
Login:



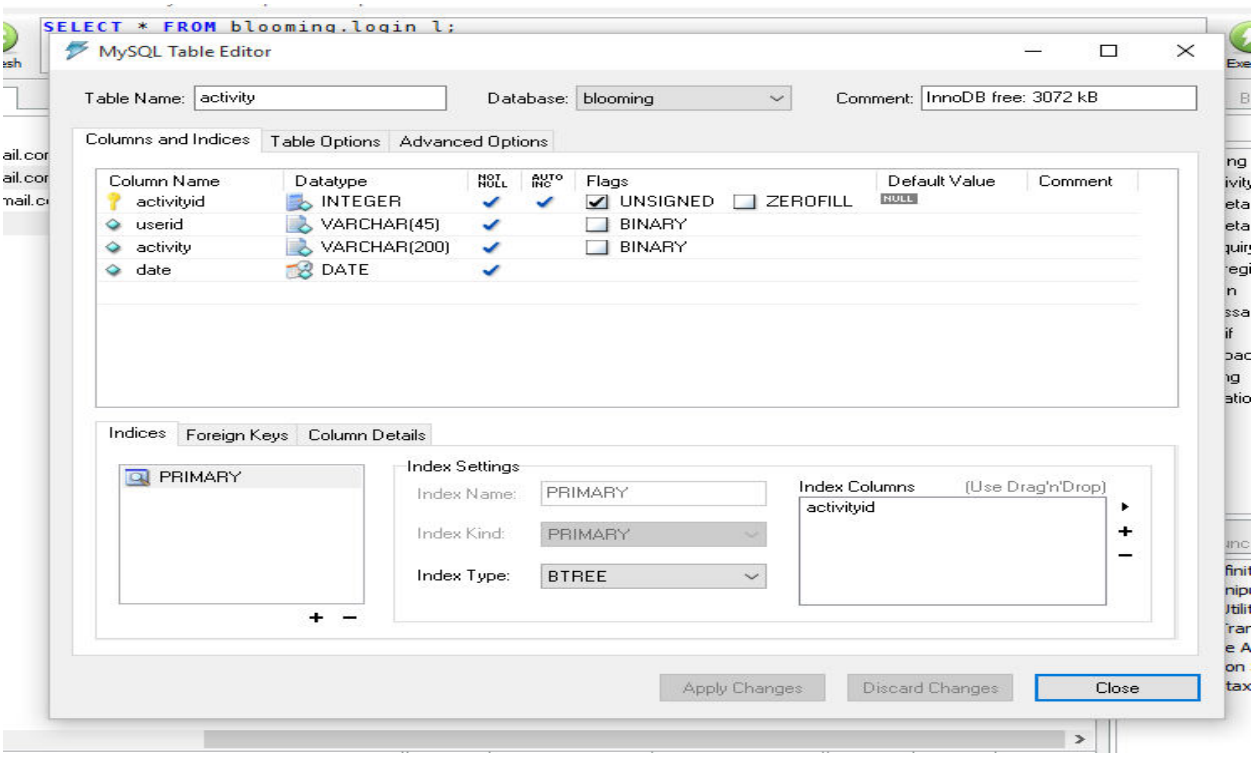
CareTaker



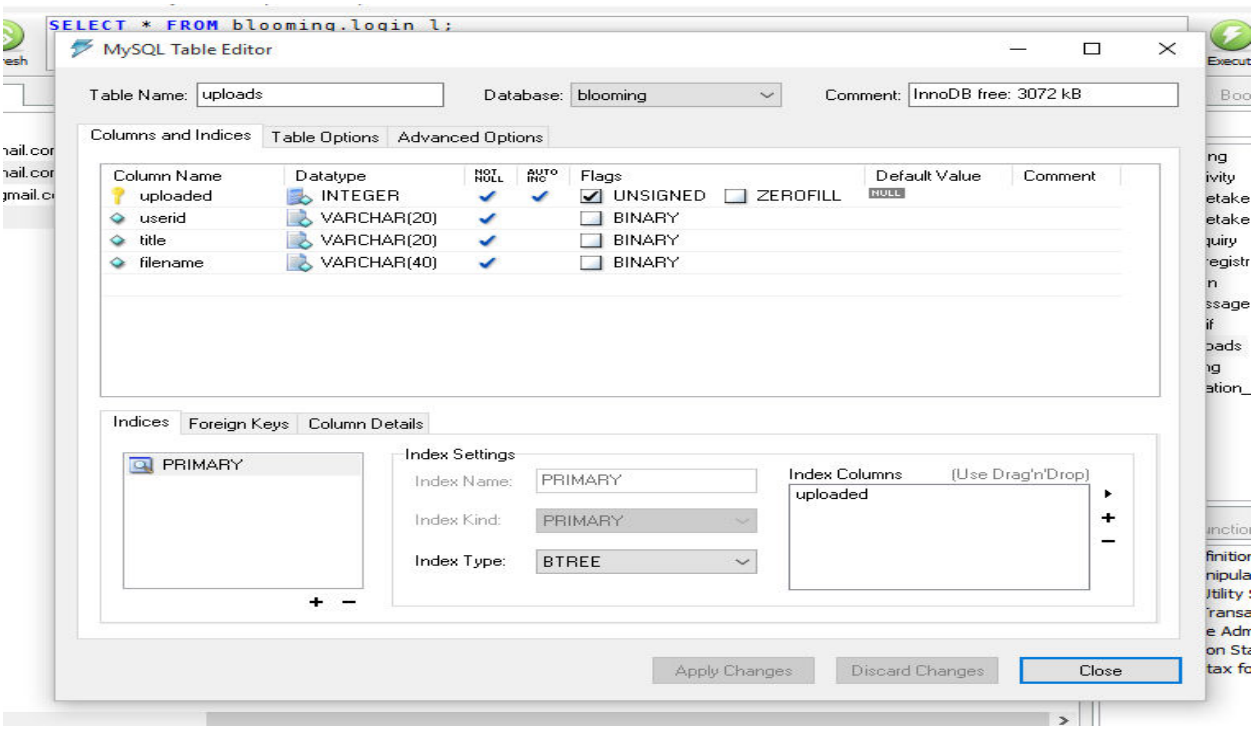
Kids Registration:



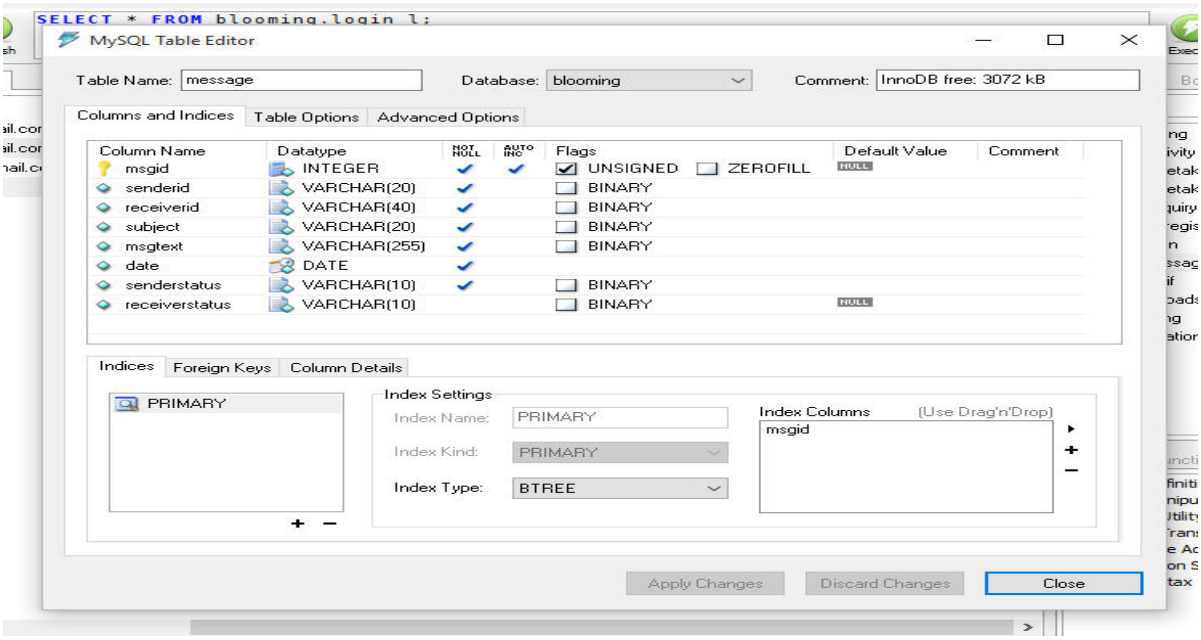
CareTakerAssignment:



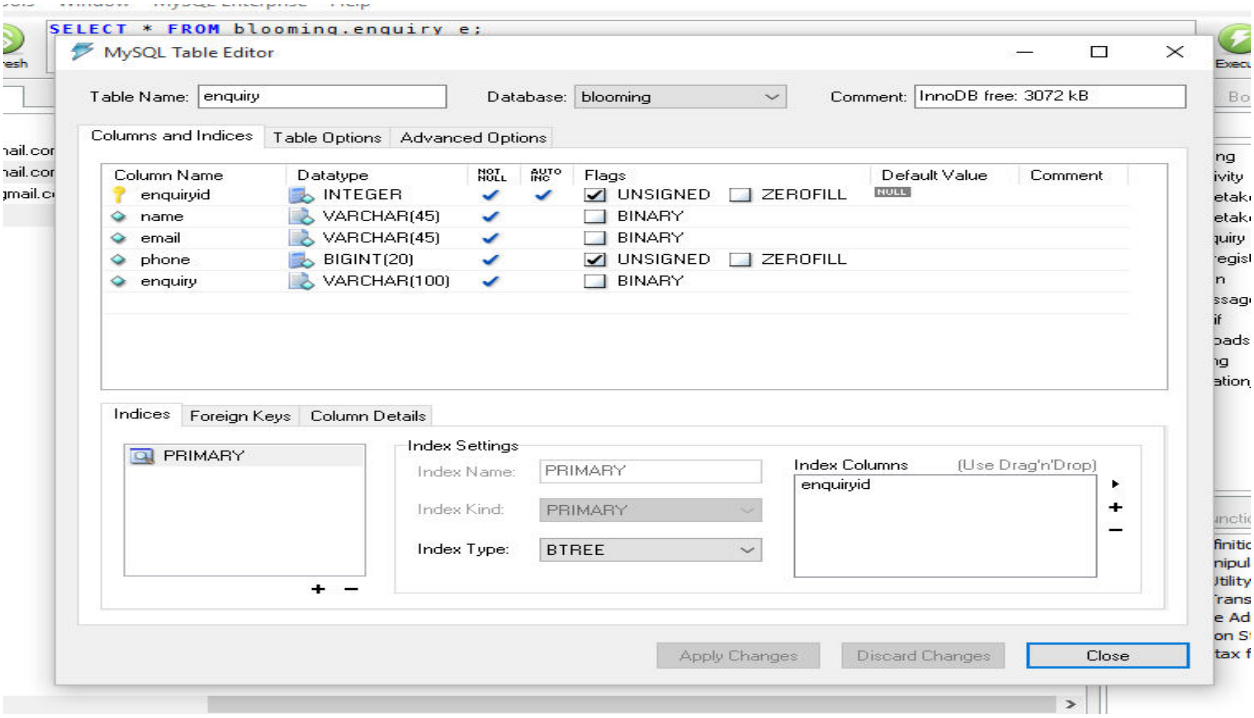
Uploads:



Message:

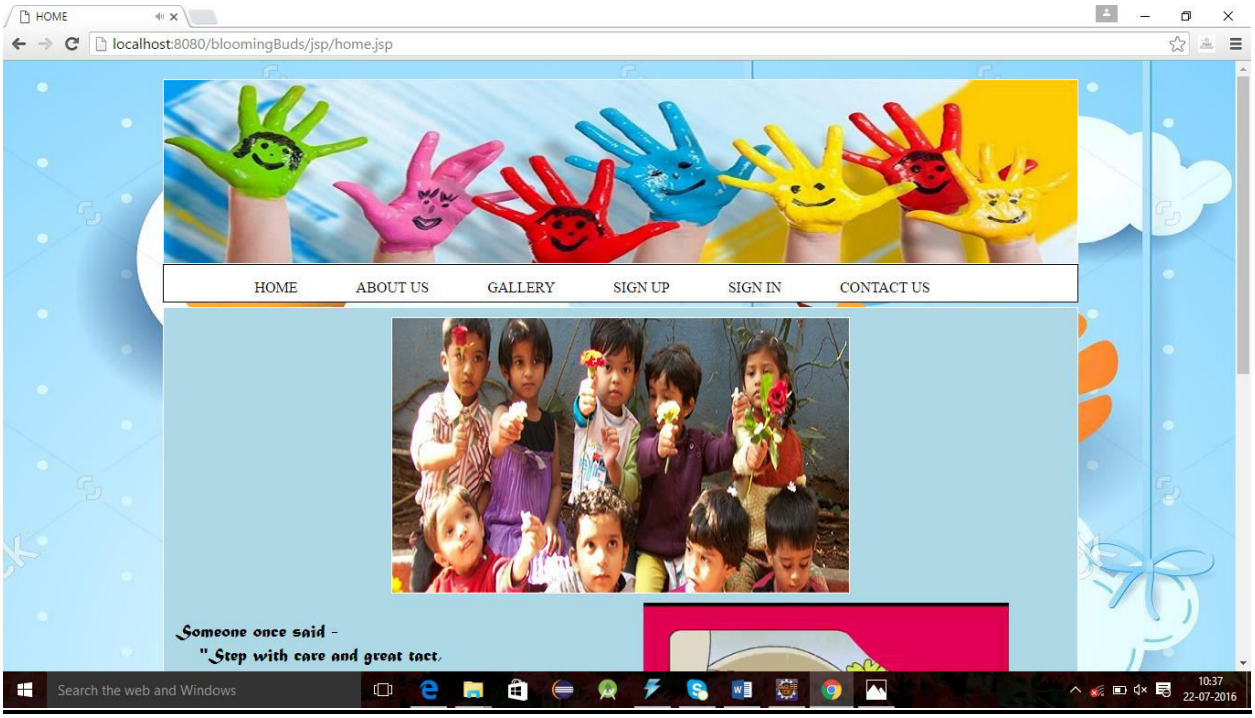

SELECT * FROM blooming.login l;
MySQL Table Editor
Table Name: message
Database: blooming
Comment: InnoDB free: 3072 kB
Columns and Indices
Table Options
Advanced Options
Column Name
Datatype
NOT NULL
AUTO INC
Flags
Default Value
Comment
msgid
INTEGER
UNSIGNED
ZEROFILL
NULL
senderid
VARCHAR(20)
BINARY
receiverid
VARCHAR(40)
BINARY
subject
VARCHAR(20)
BINARY
msgtext
VARCHAR(255)
BINARY
date
DATE
senderstatus
VARCHAR(10)
BINARY
receiverstatus
VARCHAR(10)
BINARY
NULL
Indices
Foreign Keys
Column Details
PRIMARY
Index Settings
Index Name: PRIMARY
Index Kind: PRIMARY
Index Type: BTREE
Index Columns
(Use Drag'n'Drop)
msgid
Apply Changes
Discard Changes
Close

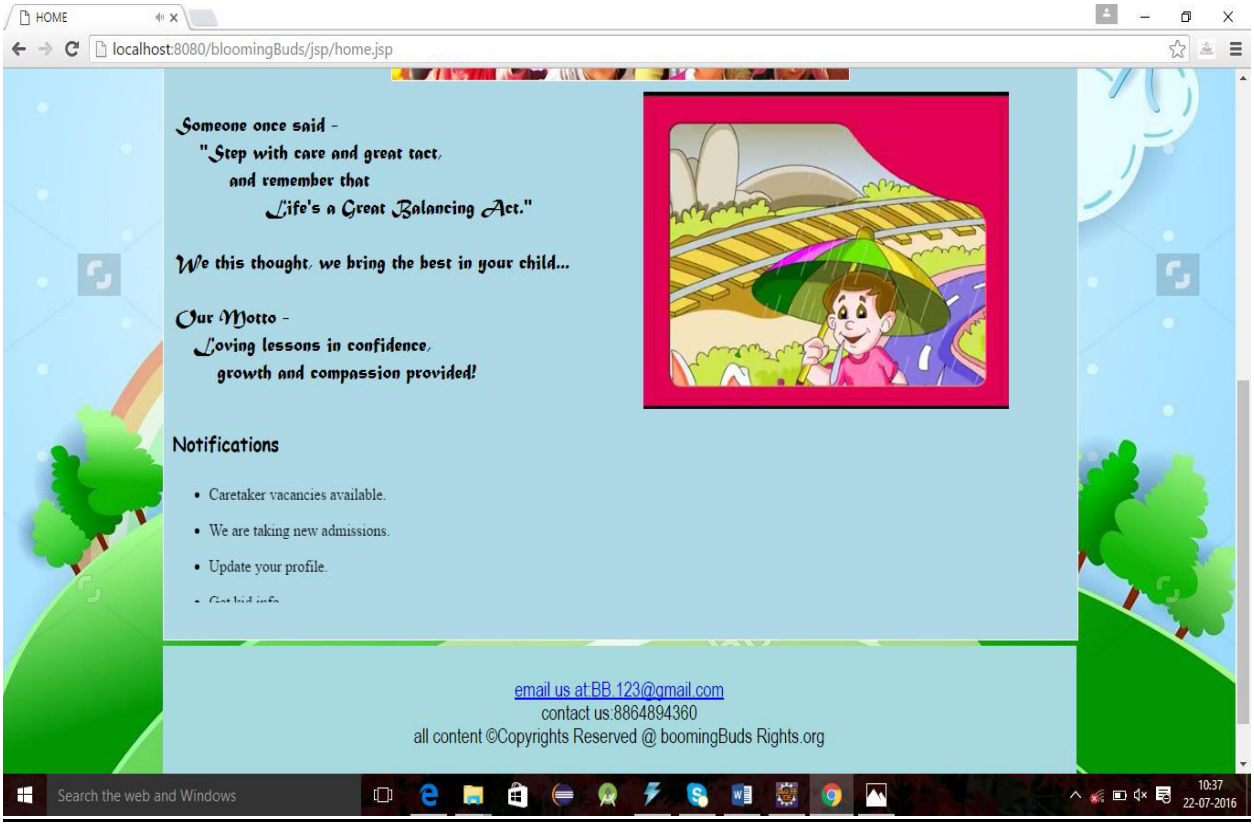
Enquiry:


SELECT * FROM blooming.enquiry e;
MySQL Table Editor
Table Name: enquiry
Database: blooming
Comment: InnoDB free: 3072 kB
Columns and Indices
Table Options
Advanced Options
Column Name
Datatype
NOT NULL
AUTO INC
Flags
Default Value
Comment
enquiryid
INTEGER
UNSIGNED
ZEROFILL
NULL
name
VARCHAR(45)
BINARY
email
VARCHAR(45)
BINARY
phone
BIGINT(20)
UNSIGNED
ZEROFILL
enquiry
VARCHAR(100)
BINARY
Indices
Foreign Keys
Column Details
PRIMARY
Index Settings
Index Name: PRIMARY
Index Kind: PRIMARY
Index Type: BTREE
Index Columns
(Use Drag'n'Drop)
enquiryid
Apply Changes
Discard Changes
Close

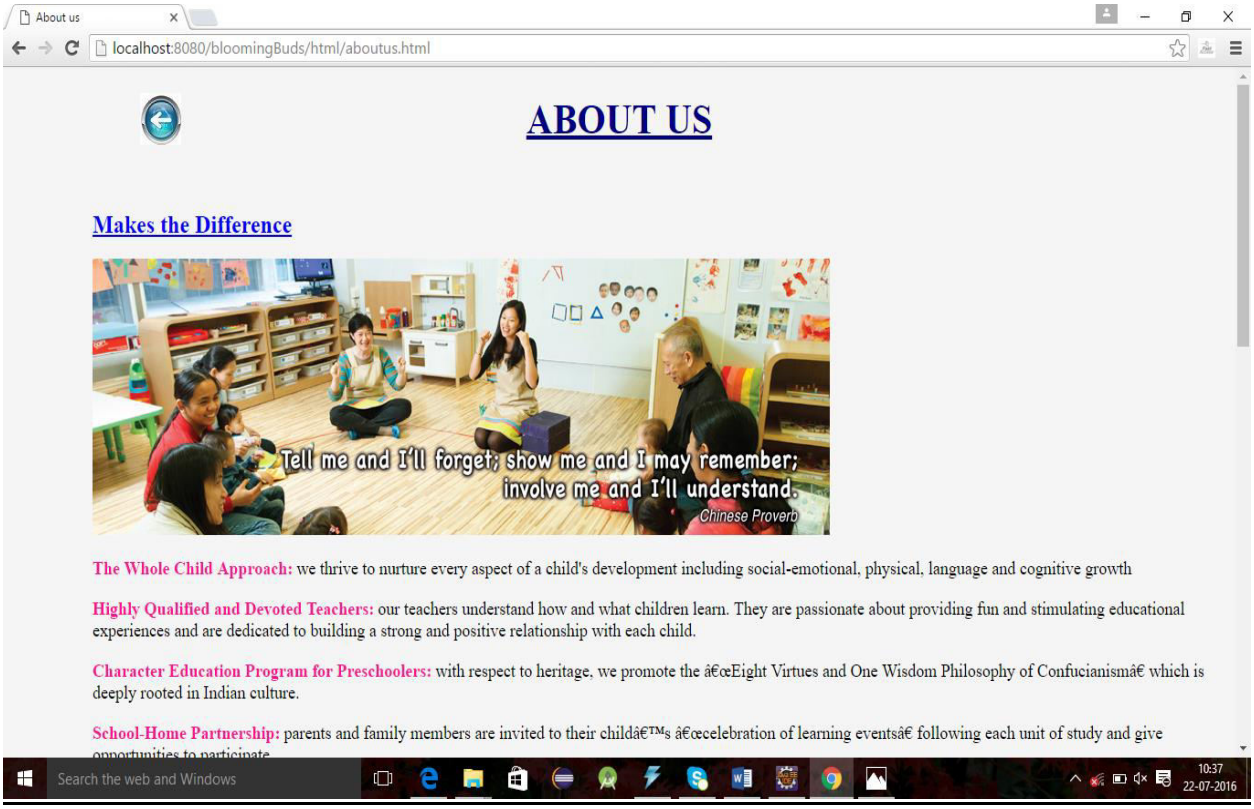
5.Screenshots

Home

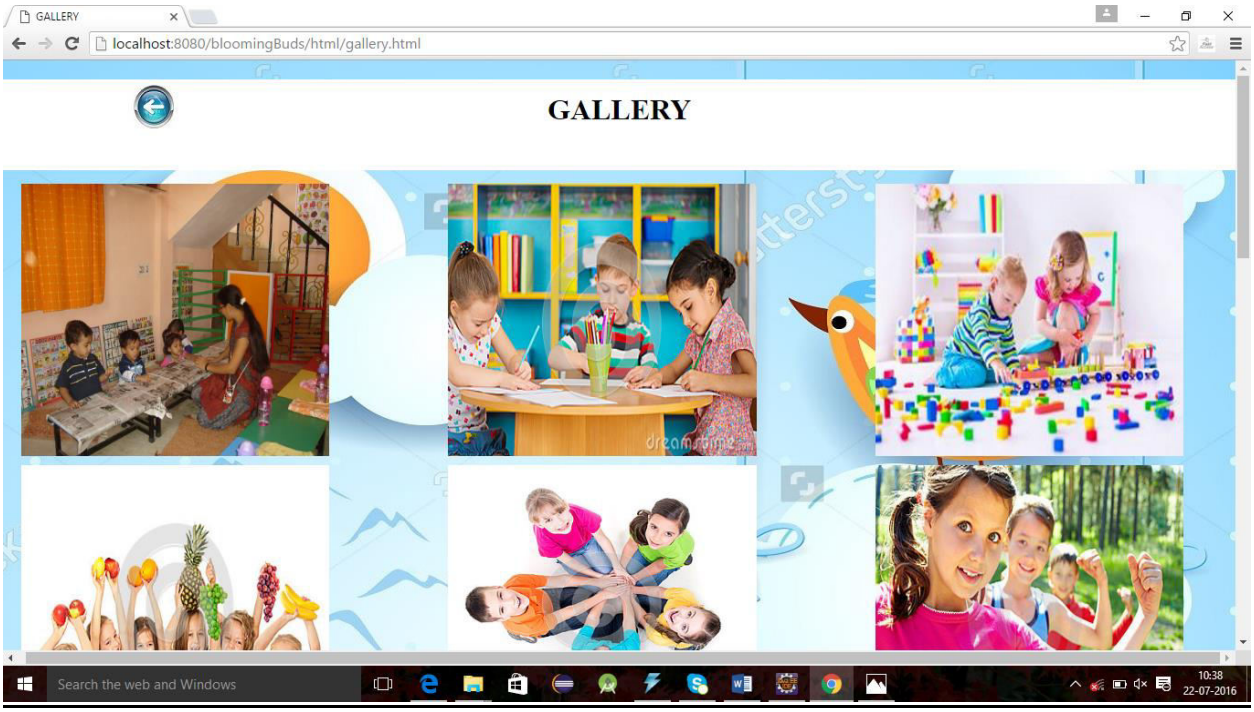




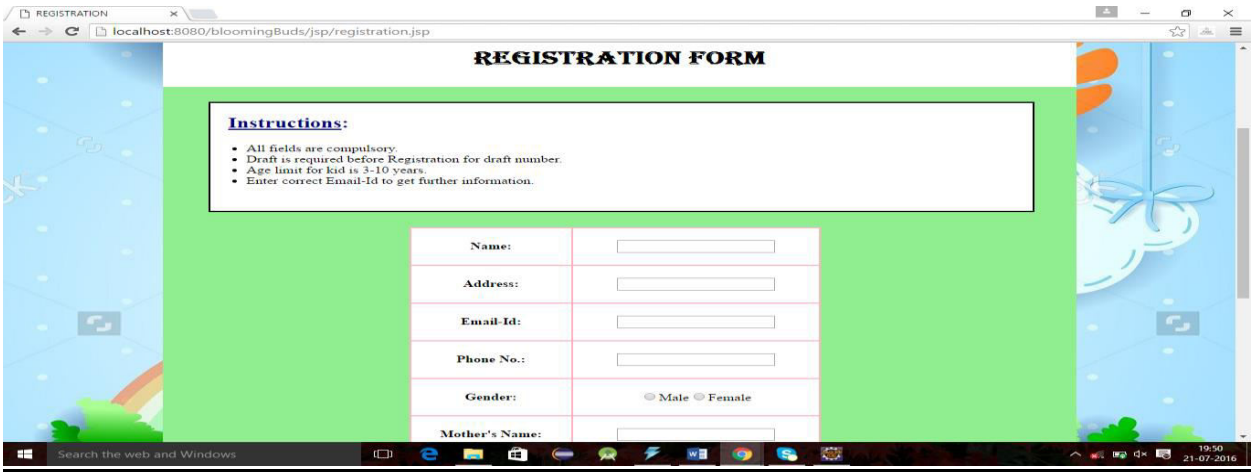
About Us



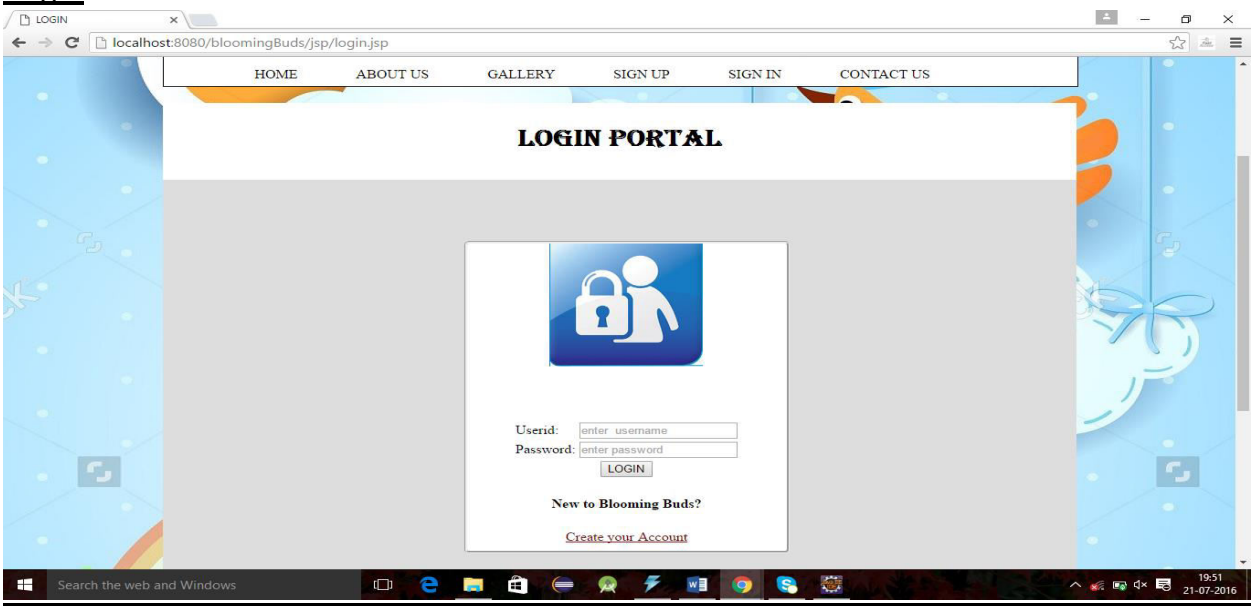
Gallery



Sign Up



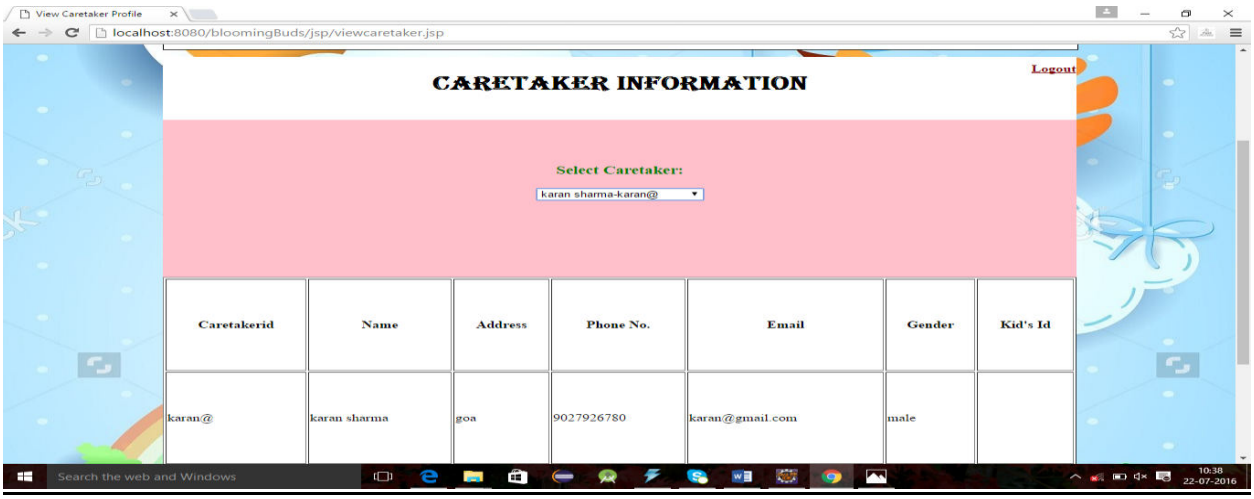
Login



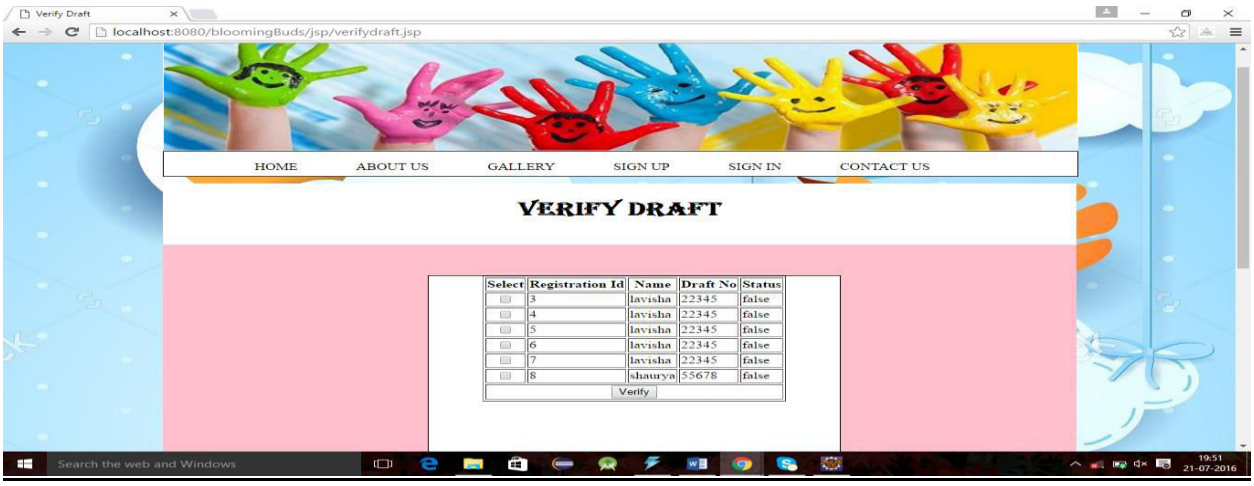
Special functionality of each panel:

Admin Panel

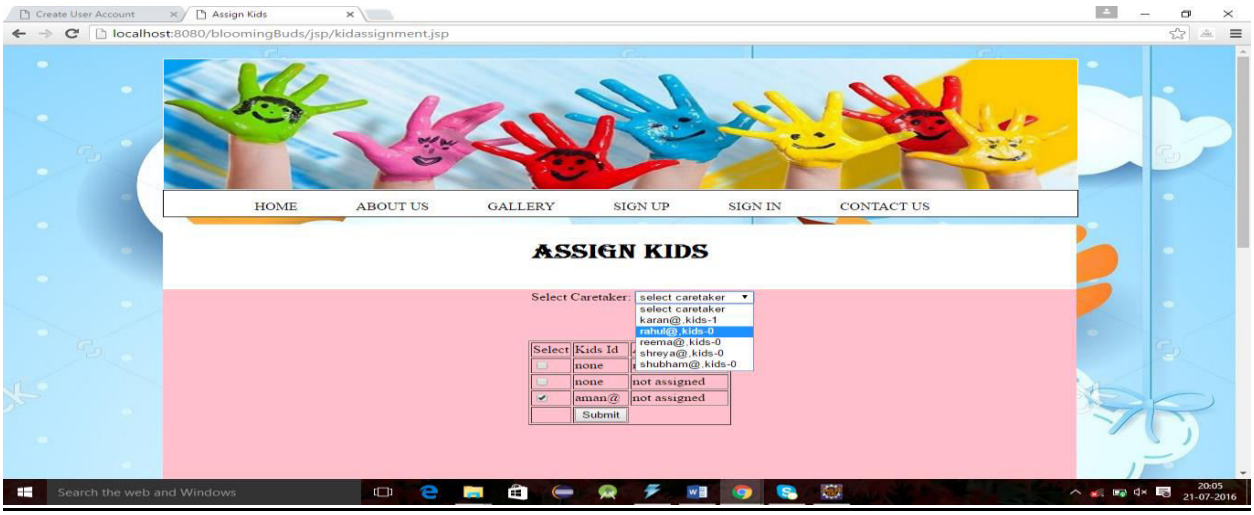
Viewing Kid's and caretaker's information



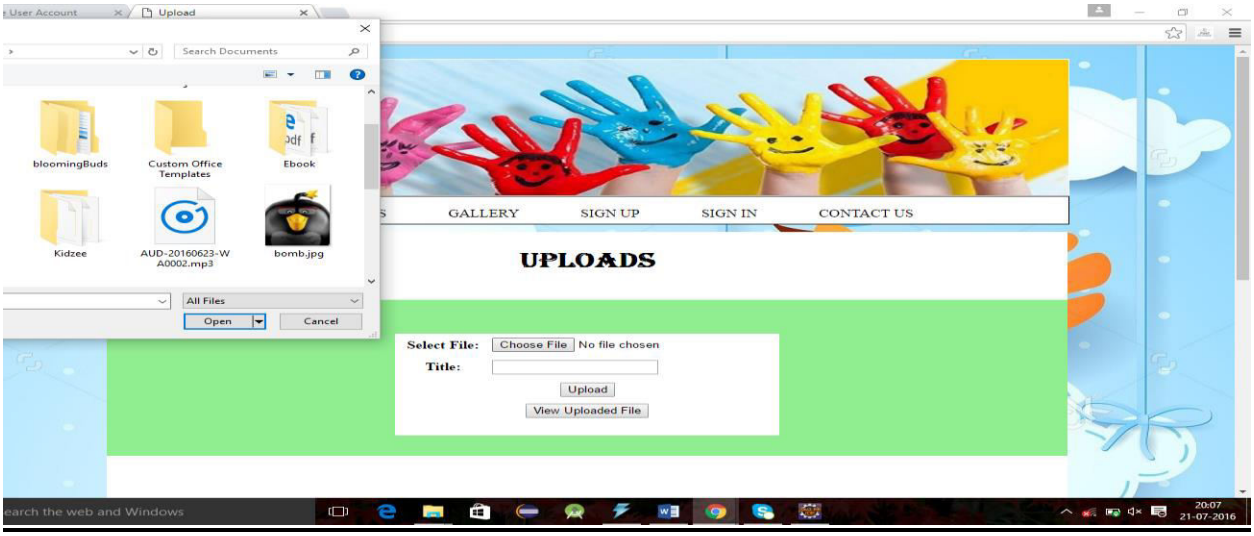
Verifying draft



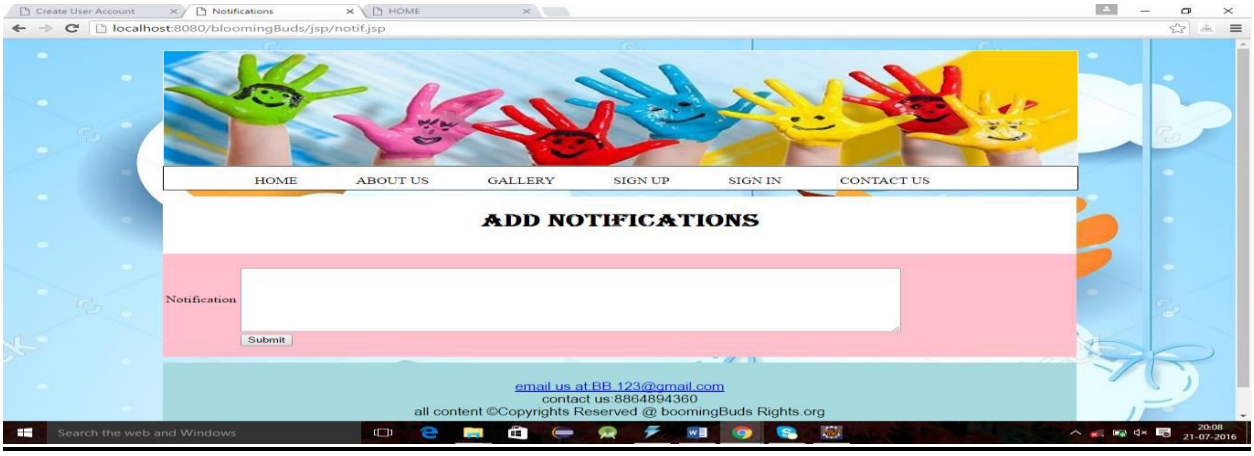
Assignment of kids to caretakers



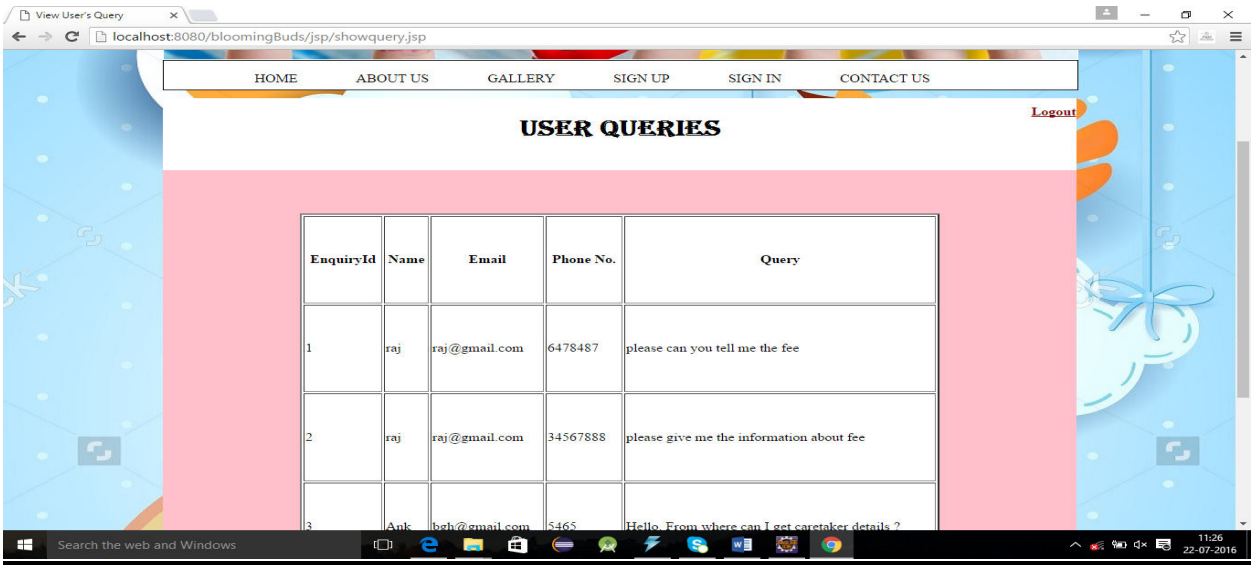
Uploading



Updating notifications

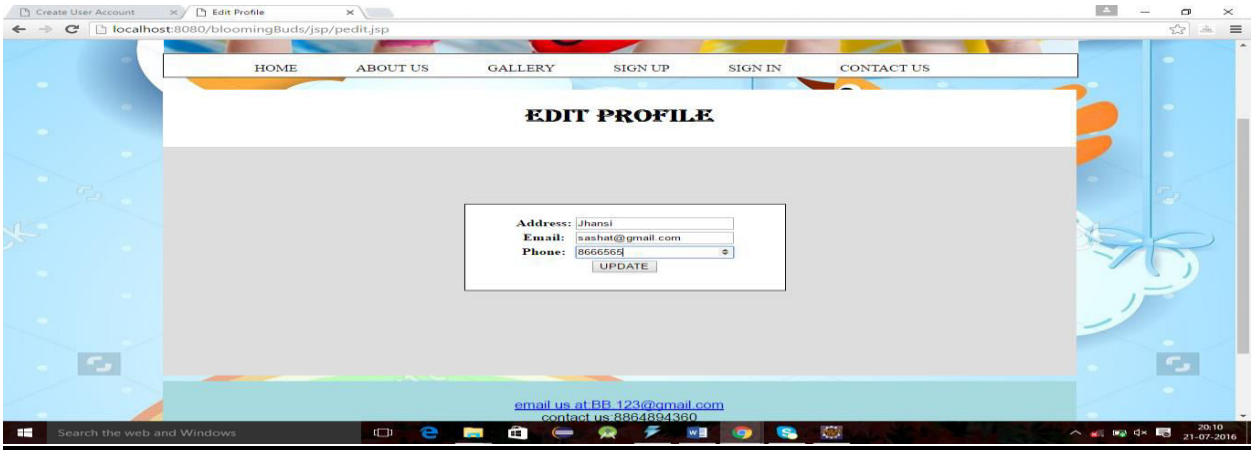


Viewing Queries



User's Panel

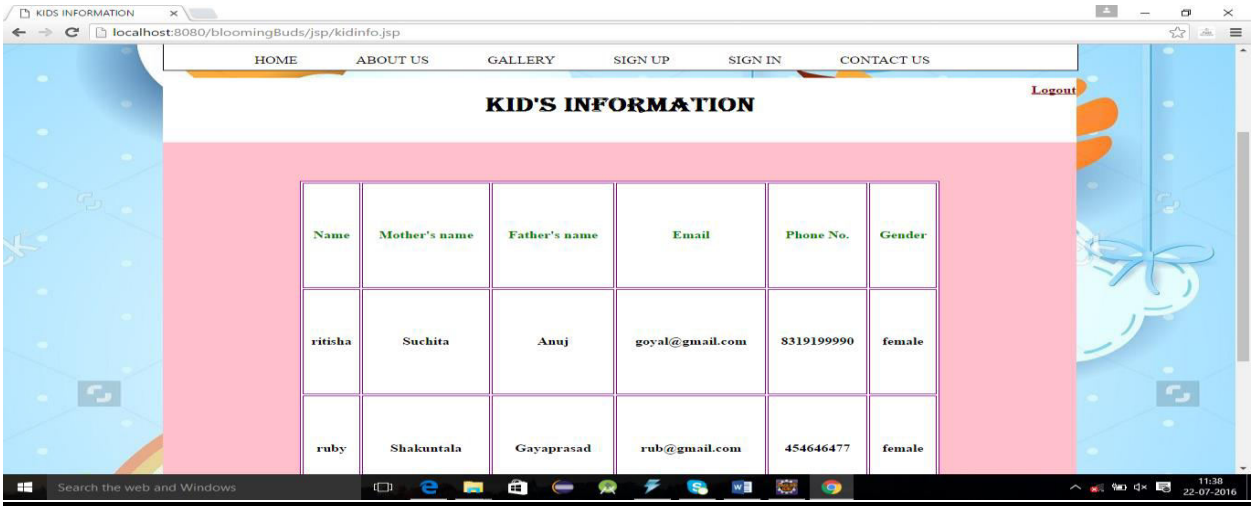
Editing profile



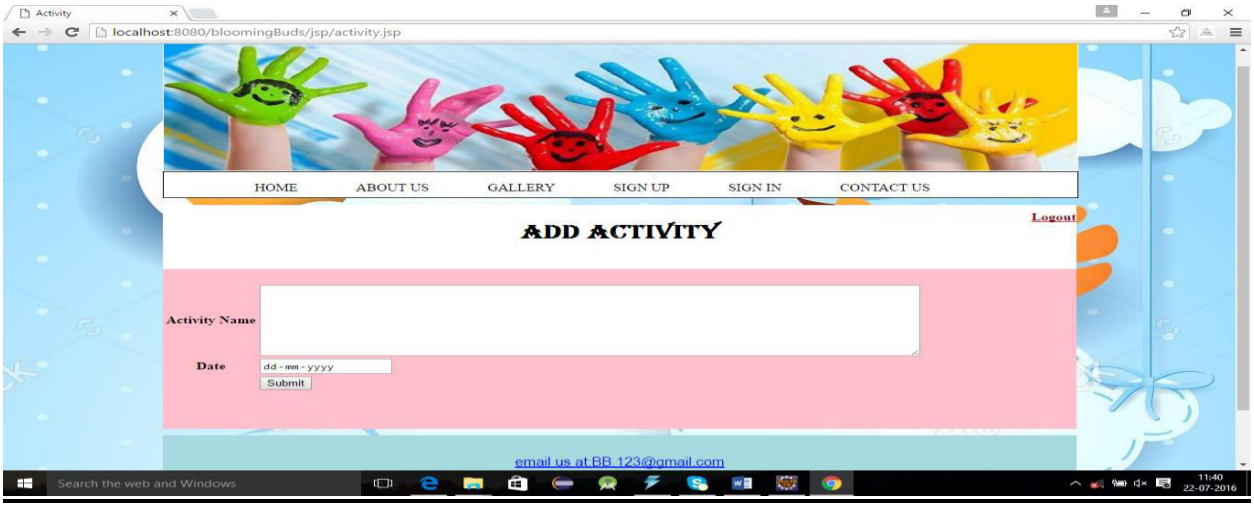
Viewing Activies

Caretaker's Panel

Viewing kids assigned to him/her

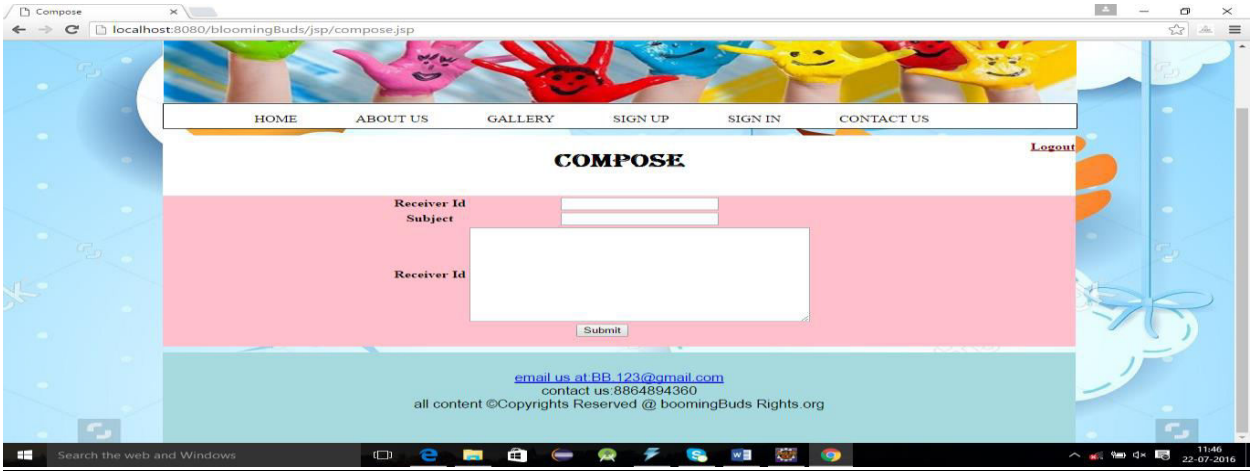


Adding activities for kids

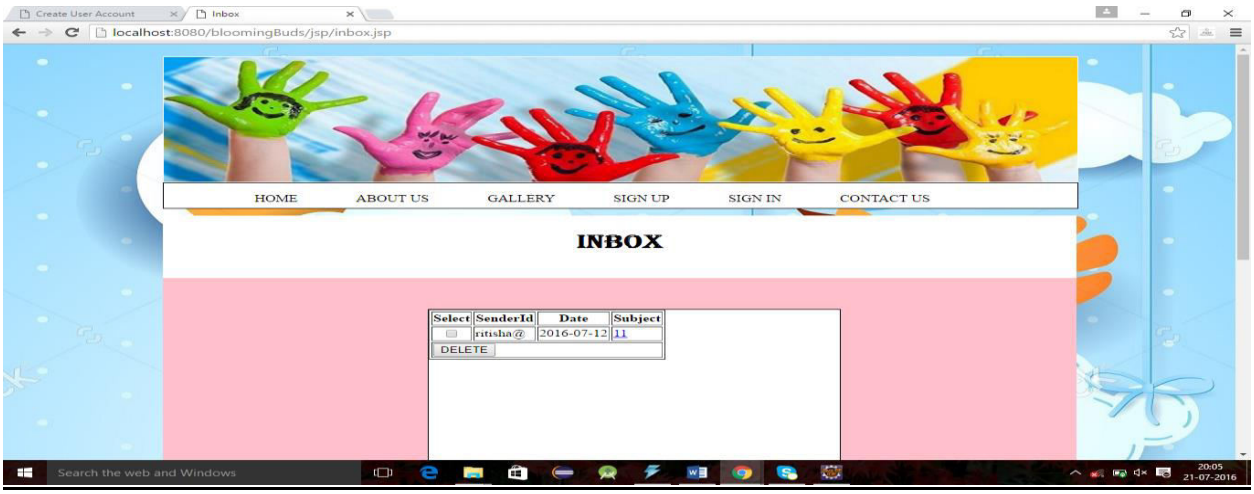


Messaging Panel:

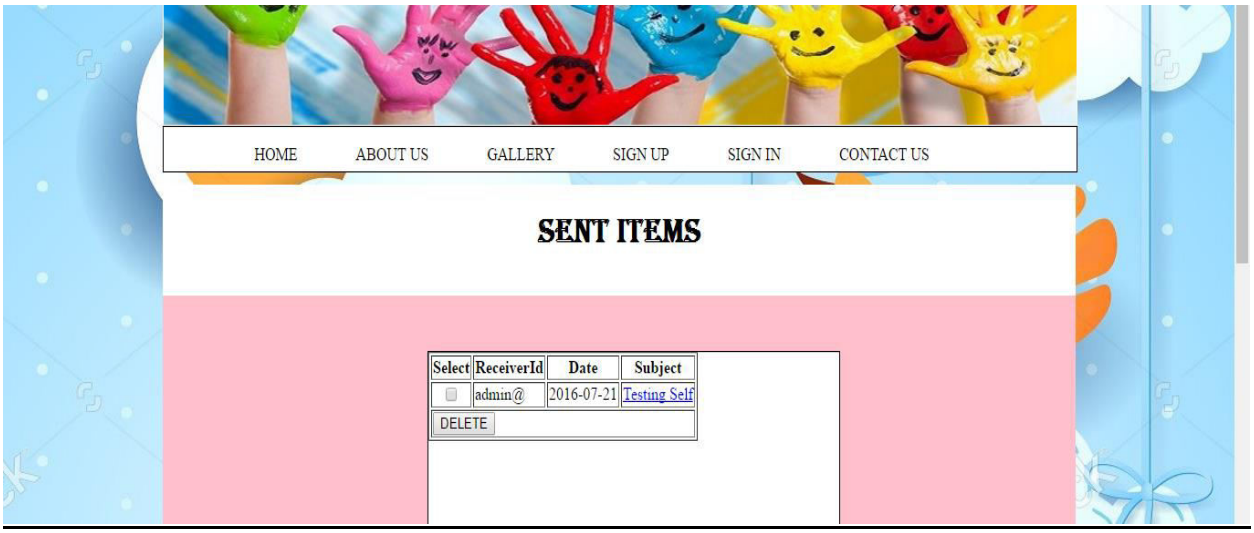
Compose mail



Inbox


Create User Account
Inbox
localhost:8080/bloomingBuds/jsp/inbox.jsp
HOME
ABOUT US
GALLERY
SIGN UP
SIGN IN
CONTACT US
INBOX
Select
SenderId
Date
Subject
ritisha@
2016-07-12
11
DELETE
Search the web and Windows
20:05
21-07-2016

Sent Items


HOME
ABOUT US
GALLERY
SIGN UP
SIGN IN
CONTACT US
SENT ITEMS
Select
ReceiverId
Date
Subject
admin@
2016-07-21
Testing Self
DELETE

6.TESTING

One of the main requirements of the software development is to develop bug free reliable system. During the process of development of any application there is always a possibility of some errors being developed. And also the software may not be as user asked for . Thus ,it becomes imperative that ,the software should be checked at every step. This helps in a bug free reliable system. If there is any bug or alteration that have to be made that can be done at the preliminarily level as the application gets more complicated after every step.

1. The objective of the testing is to discover bugs of the application. To fulfill the objective ,a series of test steps such as Unit Testing, Validation Testing, System Testing .
2. Testing is a process of executing a program with the intent of finding an error.
3. A good test case is one that has a high probability of finding an as-yet undiscovered
4. error.
5. A successful test is one that uncovers as-yet undiscovered error.
6. If the testing is conducted successfully it will uncover errors in the software.

VARIOUS TESTING APPROACHES:

1. Unit Testing

Unit Testing is white-box oriented, after the source code was developed, reviewed and verified for correct syntax, unit testing was performed. Unit testing is dynamic method for verification. In this each and every module was tested with internal logic, case in isolation and the internal logic of the modules is found to be correct and all results obtained are also correct.

2.Integration Testing

Integration testing is white box as well as black box oriented. The next level of testing is Integration testing. In this testing all the modules tested in unit testing were combined into sub system and were then tested. Integration testing was performed to ensure that the modules are integrated properly and the data flow between the modules is proper.

3.System Testing

System testing is black box oriented. In the System testing the entire software was tested with the help of actual data provided by the organization. The reference documents for this purpose are the requirement document. System testing was done to ensure that the objectives desired for the system are properly achieved by the software developed. The requirement of information useful for the manager is in the proper formats and fulfilling all the requirements.

4.Acceptance Testing

Acceptance testing was performed with realistic data of the client to demonstrate that the software was working satisfactorily. Realistic data was used to ensure the client that the entire requirements are met and the internal behavior of the system is according to his/her requirement

7. CONCLUSION

As evidence of the success of this mission, there are millions of items listed each day in thousands of different categories. There are items for almost any interest that one could imagine, from sheet music to automobiles to hand tools to real estate. And the variety doesn't stop there.

Need a computer? One may find it listed in the proper category, in any configuration from very old and obsolete to the latest greatest machine available. What about antiques? One can find an antique quilt that is up for highest bid, or maybe an old violin, whose beautiful tones have enchanted many though its years. Tickets. Maybe a ticket to the next concert of ones favorite artist or play production. One can even find that special bottle of wine, some aged, exotic cheese, and the perfect 'mood' music for that special occasion.

In this instance it may be true that on eBay, they have something for everybody, whatever their tastes may be.

8. LIMITATIONS

- single admin
- lack of responsiveness
- limited numbers of users
- limited numbers of caretaker
- it cannot be used as mobile application

9. FUTURE SCOPE

- adding new admin
- making website more interactive and responsive
- incorporating governmental interaction to make it better

10.REFERENCES

Websites

- http://www. w3schools.com
- http://www.wikipedia.com
- http://www.javatpoint.com
- http://www.tutorialspoint.com

Books:

- Core java Servlets and JSP Vol.1
- Java Web Programming