**Industry Oriented Mini Project Report**

On

**CANTEEN AUTOMATION SYSTEM**

Submitted in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology**

in

**COMPUTER SCIENCE AND ENGINEERING**

By

**G. Anusha 16241A05J4**

**J. Niveditha 16241A05J8**

**K. Anuhya 16241A05K1**

**Shaik Gulzar 16241A05M9**

Under the Esteemed guidance of

**G. LAVANYA**

**Assistant Professor**

****

**Department of Computer Science and Engineering**

##### **GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

**Bachupally, Kukatpally, Hyderabad- 500090**

**2018-2019**

##### D:\3-2 Wt\griet.jpg

##### **GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the Industry mini project entitled **"Canteen Automation System"** is submitted by

**G.Anusha 16241A05J4**

**J.Niveditha 16241A05J8**

**K.Anuhya 16241A05K1**

**Shaik Gulzar 16241A05M9**

In partial fulfillment of the requirement for the award of the degree in **BACHELOR OF TECHNOLOGY** in Computer Science and Engineering during the academic year 2018-2019.

INTERNAL GUIDE HEAD OF THE DEPARTMENT

**G.Lavanya Dr. K.Madhavi**

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

There are many people who helped us directly and indirectly to complete our project successfully. We would like to take this opportunity to thank one and all.

First of all we would like to express our deep gratitude towards our internal guide **G. Lavanya, Assistant Professor,** Department of CSE for her support in the completion of my dissertation. We wish to express our sincere thanks to **Dr.K.Madhavi, HOD, Department of CSE** and our mini project coordinator **Dr.G.Karuna** for her external support. We also thank our principal **Dr.J.Praveen** for providing the facilities to complete the dissertation.

We would like to thank all our faculty and friends for their help and constructive criticism during the project period. Finally, we are very much indebted to our parents for their moral support and encouragement to achieve our goals.

**G.Anusha**

(16421A05J4)

**J.Niveditha**

(16241A05J8)

**K. Anuhya**

(16241A05K1)

**Shaik Gulzar**

(16241A05M9)

DECLARATION

We hereby declare that the industry oriented mini project entitled **"Canteen Automation System"** is the work done during the period from 10th December 2018 to 4th April 2019 and is submitted in the partial fulfillment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad). The results embodied in this project have not been submitted to any other University or Institution for the award of any degree or diploma.

**G.Anusha**

(16421A05J4)

**J.Niveditha**

(16241A05J8)

**K.Anuhya**

(16241A05K1)

**Shaik Gulzar**

(16241A05M9)

**ABSTRACT**

**DESCRIPTION**

This project is aimed at creating a website that prominently relieves the burden on the canteen’s end as the entire method of taking order is computerised.

Within this application, all items in the order are displayed along with their equivalent options and supply details that is Items are provided by candidate. This allows canteen staff to speedily go through the orders by token id from students phone and produce the needed items.

Many students visit the canteen in their lunch break and as they have limited time to eat and return to their respective classes this software helps the students to order food and save time without waiting at canteen to order and get food. It lightens the load at canteen’s end by making the entire process oftaking orders automatic.

**CONTENTS Page No**

Certificate ii

Acknowledgement iii

Declaration iv

Abstract v

Contents vi

**1.** **Introduction 1**

1.1 Project Overview 1

1.2 Existing System 1

1.3Proposed System 2

1.3.1 Advantages of Proposed System 2

**2. System Requirements 3**

2.1 Software Requirements 3

2.2 Hardware Requirements 3

**3.** **Technology 4**

3.1HTML 4

3.2 CSS 5

3.3 MYSQL 5

3.4 XML 6

3.5 Servlet 6

3.6 JDBC Connectivity 7

**4. System design 8**

4.1 System Architecture 8

4.2 Data Flow Diagram 8

**5. Implementation 9**

5.1 Modules 9

5.1.1 Registration Module 9

5.1.2 Canteen Module 9

5.1.3 Customer Module 10

5.2 Code Snippets 10

**6. Testing 41**

6.1 System Testing 41

6.2 Types of Tests 41

**7. Screenshots 46**

7.1 Home Page 46

7.2 Login Page 46

7.3 Registration Page 47

7.4 Get Menu Page 47

7.5 Menu List Page 48

7.6 Token Page 48

7.7 Canteen End Page 49

7.8 Show Orders Page 49

7.9 Update Items Page 50

7.10 Remove Items Page 50

**8. Conclusion 51**

**9. Bibilography 52**

1. **INTRODUCTION**

**1.1 Project Overview:**

Canteen Automation System is where the students can order their food and receive food in the canteen without any delay as they can directly go and collect what they’ve ordered without waiting for a turn.

When the customer visits the ordering webpage they are presented with an up to date menu along with prices in a easy to read manner and the customer can also review what was selected before checking out.

It lightens the load at canteen’s end by making the entire process of taking orders automatic.

Once an order is placed all the details are then given to canteen’s end application which allows canteen employees to go through the orders and produce the necessary items with minimal delay and confusion.

**1.2 Existing System:**

The Challenges encountered by the manual system in canteens is efficiency and customer satisfaction.

The experience of ordering in canteens is not pleasant for customers as they have to wait in long queues before placing the order.

When the order is placed they have to wait near the counter until the order is prepared.

As mentioned the limitations of existing systems are:

* Verbal Communication.
* Billing Errors.
* Miscommunication during peak time.
* Menus not being regularly updated.
* Furthermore, not all item on the menu list are always available.

**1.3 Proposed System:**

The proposed “Canteen Automation System” is economically feasible as the system requires very fewer time factors compared to manual system and provides fast and efficient automated environment instead of slow and error prone manual system, thus reducing both time and manpower.

The system will have GUI interface and very less user training is required to learn it.

Our objective is to make a platform independent application wherein we use HTML to create webpage, Java(JDBC and servelts) for communication with database(MYSQL).

* + 1. **Avantages of Proposed System**

1. To order food rapidly in few steps
2. To make it convenient for people who have limited time
3. Cost reduction
4. Reduced paper work
5. Computerized Order
6. **SYSTEM REQUIREMENTS**

**2.1 Software Requirements:**

Supported Operating System:

* Windows 7(32 or 64 bit), Windows 8(32 or 64 bit), Windows 10(32 or 64 bit)

Supported Development Environment:

* Java(JDBC)
* Servlet
* Tomcat
* CSS
* HTML

**2.2 Hardware Requirements:**

* Processor: 1.2 GHZ
* RAM: 8 GB
* ROM: 300 GB
* Internet Connection
* Input Devices: Keyboard, mouse , hand held devices
* Output Devices: Monitor, printer , hand held devices

1. **TECHNOLOGY**

The design of the site has been done using the following technologies:-

1. HTML
2. CSS
3. MySQL
4. XML
5. SERVLET
6. JDBC CONNECTIVITY

**3.1 HTML: HYPER TEXT MARKUP LANGUAGE**

In computing, Hypertext Markup Language (HTML) is a markup language designed for the creation of web pages with hypertext and other information to be displayed in a web browser. HTML is used to structure information denoting certain text as headings, paragraphs, lists and so on and can be used to describe, to some degree, the appearance and semantics of a document. HTML’s grammar structure is the HTML DTD that was created using SGML syntax. The HTML document format is used on the Web. Web pages are built with HTML tags (codes)

embedded in the text. HTML defines the page layout, fonts, and graphic element as well as the hypertext links to other documents on the web. Each link contains the URL, or address, of a Web page residing on the

same server or any server worldwide, hence “World Wide Web”. HTML 2.0 was defined by the Internet Engineering Task Force (IETF) with a basic set of features, including interactive forms capability. Subsequent versions added more features such as blinking text, custom backgrounds and tables of contents.

**3.2 CSS: CASCADING STYLE SHEETS**

Cascading Style Sheets (CSS) is a stylesheet language used for describing the look and formatting of a document written in a markup language. While most often used to change the style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any kind of XML document, including plain XML, SVG, and XUL. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging Webpages, user interfaces for web applications, and user interfaces

for many mobile applications.

**3.3 MySQL**

Modern day websites seem to be relying more and more on complex database systems. These systems store all of their critical data and allow for easy maintenance in some cases. The Structured Query Language (SQL) is a very popular database language, and its standardization makes it quite easy to store, update and access data. One of the most powerful SQL servers out there is called MySQL and surprisingly enough, it’s free. Some of the features of MySQL Include: Handles large databases, in the area of 50,000,000+ records. No memory leaks. Tested with a commercial memory leakage detector (purify). A privilege and password system which is very flexible and secure, and which allows host-based verification. Passwords are secure since all password traffic when connecting to a server is encrypted.

**3.4 XML**

XML is a software and hardware independent tool for storing and transporting data.XML stands for eXtensible Markup Language.XML was designed to store and transport data.it was designed to be self-descriptive.XML is a W3C recommendation.XML is a quite self descriptive because it has sender information.it has receiver information, it has a heading, it has a message body. It is just information wrapped in tags.XML was designed to carry data with focus on what data it is.XML language has no predefined tags with XML, it must define both the tags and the document structure.XML is extensible language. Most XML applications will work as expected even if new data is

added or removed.

**3.5 SERVLET**

SERVLETS are java programs that runs on the java-enabled web server or application server. They are used to handle the request obtained from the web server, process the request, produce the response, then send response back to the web server. Today we are all aware of the need of creating dynamic web pages .i.e, the ones which have the capability to change the site contents according to the time or are able to generate the contents according to the request received by the client. If you like coding in JAVA ,then you will be happy to know that java there also exists a way to generate dynamic web pages and that way is Java Servlet.

**Properties of servlets:**

(1) Servlets work on the server side.

(2) Servlets capable to handling complex request obtained from web server.

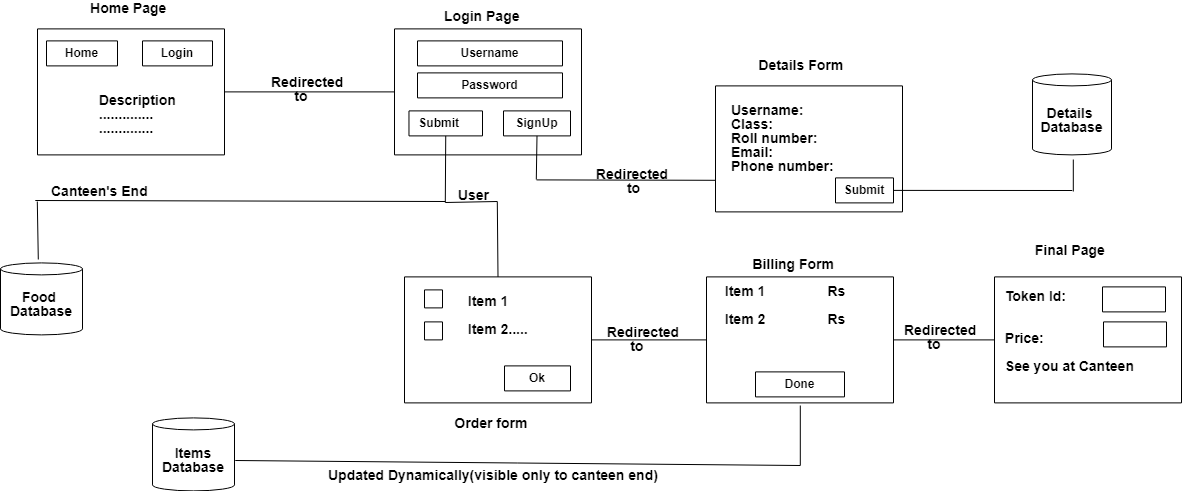
**3.6 JDBC CONNECTIVITY**

Java Database Connectivity(JDBC) is an application programming interface(API) for the programming language which defines how a client may access a database.it is a Java-based data access technology used for Java database connectivity.it is part of the Java standard edition platform, from oracle corporation.it provides

methods to query and update data in a database, and is oriented towards relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC accessible data source in Java virtual machine(JVM)host environment.

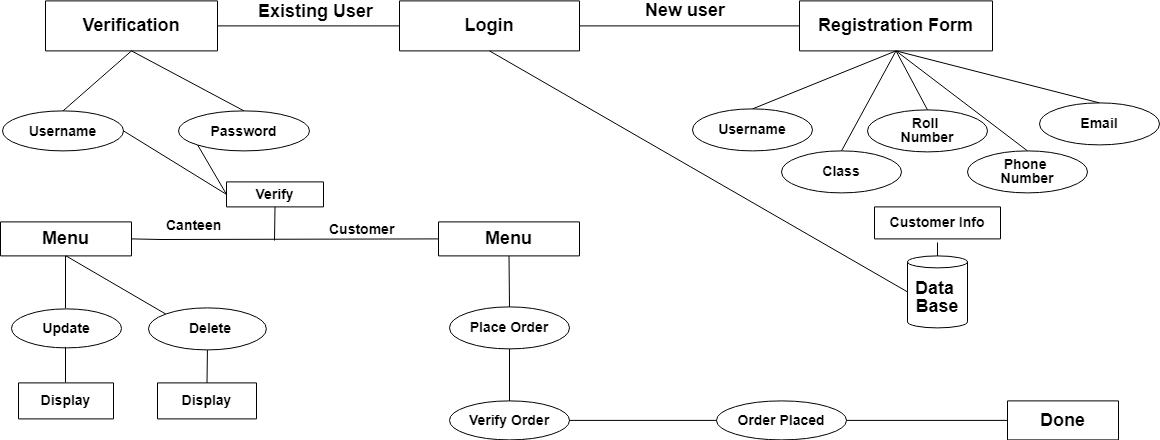
**4.SYSTEM DESIGN**

**4.1SYSTEM ARCHITECTURE:**



**Fig 4.1: System Architecture**

* 1. **Data Flow Diagram / Flow Chart**



**5.IMPLEMENTATION**

**5.1 MODULES:**

1. Registration module

2. Canteen module

3. Customer module

**5.1.1 Registration Module:**

In registration module we focus on creating a home page which has the description of the website and a login button. Upon clicking onto login the user is redirected into a login page where an existing users can signin into their account and also a registration link is provided for new users to signup. Upon clicking onto SignIn page the users are redirected to either canteen or customer module based on the username and password provided. Upon clicking onto SignUp or registration link the user is redirected to a registration form where he/she can register themselves and then login into customer module.

**5.2.2 Canteen Module :**

The canteen module is accessible to only the canteen end. They can be able to see the items present on the menu and whether the items are available or not available. The canteen admin is also provided three buttons – Remove , Update , ShowOrders.

In Remove upon entering the text item and the cost the item can be deleted from the menu. In Update upon entering the text item and cost item and its availability the item in the menu list is changed accordingly. The ShowOrders option show the canteen what orders the customers have made. And upon delivering the orders the order can be deleted.

**5.3.3 Customer Module:**

In the customer module when the users login they are shown the menu with items and their cost and a checkbox. The users can selected the items and upon clicking next they are shown the total cost and their token number to collect their items when they reach the canteen.

**5.2 Code Snippets:**

**//Canteenservlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.sql.\*;

import java.io.\*;

public class CanteenServlet extends HttpServlet

{

Public void doPost(HttpServletRequest request,HttpServletResponse response)throws ServletException, IOException

{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

try{

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

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

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("SELECT \* FROM Food\_TDB");

ResultSetMetaData rsmd=rs.getMetaData();

int nc=rsmd.getColumnCount();

out.print("<body text=black background='LIB/image/peach.png' >");

out.print("<h1 align=center>ORDER</h1><hr><br><br>");

out.print("<table align=center width=100% border=0 cellspacing=0 cellpadding=5>");

out.print("<tr><th>ITEMS</th><th>COST IN RS:</th><th>AVAILABILITY</th></tr>");

out.println("<style>");

out.print("h1 {");

out.print("font-size:13pt;");

out.print("padding:2px;");

out.print("border: 3px solid black;");

out.print("}");

out.print("</style>");

while(rs.next())

{

out.print("<tr>");

for(int c=1;c<=nc;c++)

{

out.print("<td align=center>"+rs.getString(c)+"</td>"); }

out.print("</tr>");

}

out.print("<tr>");

out.print("<td align=center><form action=remove.html>");

out.print("<input type=submit value=Remove size=25></form></td>");

out.print("<td align=center><form action=update.html>");

out.print(" <input type=submit value=Update></form></td>");

out.print("<td align=center><form action=orders.com>");

out.print(" <input type=submit value=ShowOrders></form></td>");

out.print("</tr>");

out.print("</table>");

st.close();

on.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

{

doPost(request,response);

}//doGet

}//class

**//DoneServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.\*;

import java.util.Scanner;

public class DoneServlet extends HttpServlet

{

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

{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

ResultSet rs=null;

String dvalues[]=request.getParameterValues("checked");

try{

Class.forName

("oracle.jdbc.driver.OracleDriver");

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

Statement st=con.createStatement();

rs=st.executeQuery("SELECT \* FROM order\_db");

ResultSetMetaData rsmd=rs.getMetaData();

int nc=rsmd.getColumnCount();

for(int i=0;i<dvalues.length;i++)

{

CallableStatement cst=con.prepareCall("{call ldelete(?)}");

cst.setInt(1,Integer.parseInt(dvalues[i]));

cst.execute();

out.print("done");

}

RequestDispatcher rdc=request.getRequestDispatcher("/canteen.com");

rdc.forward(request,response);

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

{

doPost(request,response);

}//doGet

}//class

**//FoodServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.\*;

import java.util.Scanner;

public class FoodServlet extends HttpServlet

{

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

{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

try{

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

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

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("SELECT \* FROM Food\_TDB");

ResultSetMetaData rsmd=rs.getMetaData();

int nc=rsmd.getColumnCount();

//Response On Browser

out.print("<body background='LIB/image/peach.png'>");

out.print("<h1 align=center>MENU</h1><hr><br><br>");

out.print("<form action=next.com method=post><br><br>");

out.print("<table align=center width=100% border=0 cellspacing=0 cellpadding=5>");

out.print("<tr><th>SELECT</th><th>ITEM</th><th>COST IN RS.</th></tr>");

out.println("<style>");

out.print("h1 {");

out.print("font-size:13pt;");

out.print("padding:2px;");

out.print("border: 3px solid black;");

out.print("}");

out.print("</style>");

while(rs.next())

{

out.print("<tr>");

if(rs.getString(3).equals("a"))

out.print("<td align=center><input type=checkbox name=selected value="+rs.getString(1)+">");

for(int c=1;c<nc;c++)

{

if(rs.getString(3).equals("a"))

out.print("<td align=center><b>"+rs.getString(c)+"</b></td>");

}

out.print("</tr>");

}

out.print("<tr>");

out.print("<td></td>");

out.print("<td align=center><input type=submit value=NEXT></td>");

out.print("<td></td>");

out.print("</tr>");

out.print("</table>");

out.print("</form>");

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

doPost(request,response);

}//doGet

}//class

**//LoginValidation Servlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.\*;

import java.util.Scanner;

public class LoginValidation extends HttpServlet

{

public void service(ServletRequest request,

ServletResponse response)throws ServletException,IOException{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

String username= request.getParameter("username");

String pword= request.getParameter("upassword");

Connection con=null;

Statement st=null;

try

{

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

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

st=con.createStatement();

ResultSet rs=st.executeQuery("select \* from Login\_DB");

out.print(""+username+""+pword);

if(username.equalsIgnoreCase("admin")&&pword.equals("canteen"))

{

RequestDispatcher rdc=request.getRequestDispatcher("/canteen.com");

rdc.forward(request,response);

}

while(rs.next())

{

if(rs.getString(1).equalsIgnoreCase(username) && rs.getString(2).equals(pword))

{

RequestDispatcher rdf=request.getRequestDispatcher("/food.html");

rdf.forward(request,response);

}

else

{}

}

RequestDispatcher rdl=request.getRequestDispatcher("/login.html");

rdl.forward(request,response);}

catch (Exception e)

{

e.printStackTrace();

}

finally

{

try

{

if(st!=null)

st.close();

}

catch (SQLException s)

{

s.printStackTrace();

}

try

{

if(con!=null)

con.close();

}

catch (SQLException e)

{

e.printStackTrace();

}

}

out.close();

}

}

**//NextServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.sql.\*;

import java.io.\*;

import java.util.\*;

public class NextServlet extends HttpServlet{

Random rmno=new Random();

Public void doPost(HttpServletRequest request,HttpServletResponse response)throws ServletException, IOException{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

//Req.p.Logic (Read Table name)

String fvalues[]=

request.getParameterValues("selected");

String lfvalues[]=new String[5];

int sum=0;

//Response On Browser

out.print("<body bgcolor=blue>");

try{

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

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

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("SELECT \* FROM Food\_tDB");

ResultSetMetaData rsmd=rs.getMetaData();

int nc=rsmd.getColumnCount();

//Response On Browser

out.print("<body text=black background='LIB/image/peach.png' >");

out.print("<h1 align=center>ORDER</h1><hr><br><br>");

out.print("<form action=next.com method=post><br><br>");

out.print("<table align=center width=100% border=0 cellspacing=0 cellpadding=5>");

out.print("<tr><th><h3>ITEMS SELECTED</th><th>COST IN RS:</h3></th></tr>");

out.println("<style>");

out.print("h1 {");

out.print("font-size:13pt;");

out.print("padding:2px;");

out.print("border: 3px solid black;");

out.print("}");

out.print("</style>");

int token=rmno.nextInt();

//out.print("@hii");

while(rs.next())

{

for(int i=0;i<fvalues.length;i++){

if(((rs.getString(1)).equals(fvalues[i]))){

out.print("<tr>");

if(rs.getString(3).equals("a"))

sum+=Integer.parseInt(rs.getString(2));

for(int c=1;c<nc;c++)

{

out.print("<td align=center><h3>"+rs.getString(c)+"</h3></td>");

}

out.print("</tr>");

}

}

}

out.print("<tr>");

out.print("</tr>");

out.print("<tr>");

out.print("<center><td><h3>Amount to be paid:"+sum+"</h3></td><center>");

out.print("</tr>");

out.print("</table>");

out.print("<center><h2>Token Id:"+token+"</h2></center>");

out.print("<p align=center><center><h2>Thank You!! Have A Nice Day</h2><center></p>");

out.print("<p align=center><center><h2>See You At The Canteen</h2></center></p>");

int i;

for(int j=0;j<fvalues.length;j++)

lfvalues[j]=fvalues[j];

for(i=fvalues.length;i<5;i++)

lfvalues[i]="n";

CallableStatement cst=con.prepareCall("{call POinsert(?,?,?,?,?,?,?)}");

cst.setInt(1,token);

/\*for(i=1;i<=fvalues.length;i++)

cst.setString(i+1,fvalues[i]);

for(int j=(i+1);j<=6;j++)

cst.setString(j,"nullss");\*/

cst.setString(2,lfvalues[0]);

cst.setString(3,lfvalues[1]);

cst.setString(4,lfvalues[2]);

cst.setString(5,lfvalues[3]);

cst.setString(6,lfvalues[4]);

cst.setInt(7,sum);

cst.execute();

//cst.commit();

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

doPost(request,response);

}//doGet

}//class

**//OrderServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.sql.\*;

import java.io.\*;

public class OrderServlet extends HttpServlet

{

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

{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

//DBLogic

try{

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

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

Statement st=con.createStatement();

ResultSet rs=

st.executeQuery("SELECT \* FROM Order\_Db");

ResultSetMetaData rsmd=

rs.getMetaData();

int nc=rsmd.getColumnCount();

//Response On Browser

out.print("<body text=black background='LIB/image/peach.png' >");

out.print("<h1 align=center>ORDERS</h1><hr><br><br>");

out.print("<table align=center width=100% border=0 cellspacing=0 cellpadding=5>");

out.print("<tr><th>SELECT</th><th>TOKEN</th><th>ITEM1</th><th>ITEM2</th><th>ITEM3</th><th>ITEM4</th><th>ITEM5</th><th>COST</th></tr>");

out.println("<style>");

out.print("h1 {");

out.print("font-size:13pt;");

out.print("padding:2px;");

out.print("border: 3px solid black;");

out.print("}");

out.print("</style>");

out.print(" <form action=lastdelete.com method=post>");

while(rs.next())

{

out.print("<tr>");

out.print("<td align=center><b>

<input type=checkbox name=checked value="+rs.getInt(1)+"></b>");

for(int c=1;c<=nc;c++)

{

out.print("<td align=center><b>"+rs.getString(c)+"</b></td>"); }

out.print("</tr>");

}

out.print("<tr>");

out.print("<td>");

out.print("<input type=submit value=Done!></td>");

out.print("</tr>");

out.print("</table>");

//out.print("</form>");

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

doPost(request,response);

}//doGet

}//class

**//RemoveServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.sql.\*;

import java.io.\*;

public class RemoveServlet extends HttpServlet

{

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

{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

String Itemname=request.getParameter("item");

ResultSet rs=null;

//DBLogic

try{

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

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

Statement st=con.createStatement();

rs=st.executeQuery("SELECT \* FROM Food\_TDB");

ResultSetMetaData rsmd=rs.getMetaData();

int nc=rsmd.getColumnCount();

CallableStatement cst=con.prepareCall("{call Pdelete(?)}");

//Response On Browser

out.print("<body bgcolor=blue text=white>");

while(rs.next())

{

if(rs.getString(1).equalsIgnoreCase(Itemname))

{

cst.setString(1,Itemname);

cst.execute();

}

}

RequestDispatcher rdc=request.getRequestDispatcher("/canteen.com");

rdc.forward(request,response);

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

doPost(request,response);

}//doGet

}//class

**//SignupSql Servlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.io.\*;

import java.sql.\*;

import java.util.Scanner;

public class SignupSql extends HttpServlet

{

public void service(ServletRequest request,

ServletResponse response)throws ServletException,IOException{

PrintWriter out=response.getWriter();

response.setContentType("text/html");

Connection con=null;

CallableStatement cst=null;

String userid=request.getParameter("uid");

String name=request.getParameter("Name");

String branch=request.getParameter("branch");

String email=request.getParameter("email");

int phone=Integer.parseInt(request.getParameter("mobileno"));

String suser=request.getParameter("signupusername");

String spwd=request.getParameter("signuppwd");

try

{

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

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

cst=con.prepareCall("{call Pinsert(?,?,?,?,?,?,?)}");

cst.setInt(1,2);

cst.setString(2,userid);

cst.setString(3,name);

cst.setString(4,branch);

cst.setString(5,email);

cst.setInt(6,phone);

cst.setString(7,suser);

cst.execute();

CallableStatement cst1=con.prepareCall("{call Pinsertldb(?,?)}");

cst1.setString(1,suser);

cst1.setString(2,spwd);

cst1.execute();

RequestDispatcher rd=request.getRequestDispatcher("/login.html");

rd.forward(request,response);

}

catch (Exception e)

{

e.printStackTrace();

}

finally

{

try

{

if(cst!=null)

cst.close();

}

catch (SQLException s)

{

s.printStackTrace();

}

try

{

if(con!=null)

con.close();

}

catch (SQLException e)

{

e.printStackTrace();

}

}

out.close();

}

}

/**/UpdateServlet**

import javax.servlet.http.\*;

import javax.servlet.\*;

import java.sql.\*;

import java.io.\*;

public class UpdateServlet extends HttpServlet{

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

PrintWriter out=response.getWriter();

response.setContentType("text/html");

String Itemname=request.getParameter("item");

String Available=request.getParameter("available");

ResultSet rs=null;

//DBLogic

try{

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

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

Statement st=con.createStatement();

rs=st.executeQuery("SELECT \* FROM Food\_TDB");

ResultSetMetaData rsmd=

rs.getMetaData();

int nc=rsmd.getColumnCount();

CallableStatement cst=con.prepareCall("{call PUpdate(?,?)}");

//Response On Browser

out.print("<body bgcolor=blue text=white>");

while(rs.next())

{

if(rs.getString(1).equalsIgnoreCase(Itemname))

{

cst.setString(1,Itemname);

cst.setString(2,Available);

cst.execute();

}

}

RequestDispatcher rdc=request.getRequestDispatcher("/canteen.com");

rdc.forward(request,response);

st.close();

con.close();

}//try

catch(ClassNotFoundException cnfe)

{

out.print

("<h2>Unable to process...</h2>");

}

catch(SQLException se)

{

out.print("<h2>Unable to process.."+se+"</h2>");

}

out.print("</body>");

out.close();

}//doPost

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

doPost(request,response);

}//doGet

}//class

**//login.html**

<html>

<head>

<script>

function Verify()

{

var uname,pwd;

uname=document.loginform.username.value;

pwd=document.loginform.upassword.value;

if(uname=="" || pwd==""){

alert("enter valid user name and password");

return false;

}

}

</script>

</head>

<body style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<form action="login.com" method="post" name=loginform onSubmit="return(Verify())">

<br></br><br></br>

<br></br><br></br>

<br></br><br></br>

<center><h1 style='font-family:"Courier New", Courier, monospace;'><caption>LOGIN INTO GRIET CANTEEN</caption></h1></center>

<br></br>

<table border="0" width="25%" height="15%" align="center"

style='font-family:"Courier New", Courier, monospace; font-size:150%'>

<tr>

<td >USERNAME :</td>

<td><input type="text" name="username" size="30"></td>

</tr>

<tr>

<td>PASSWORD : </td>

<td><input type="password" name="upassword" size="30"></td>

</tr>

<tr></tr><tr></tr>

<tr></tr><tr></tr>

<tr></tr><tr></tr>

<td></td>

<tr>

<td align="center"><input type="submit" value="SIGN IN" style="height:50px; width:75px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;" ></form></td>

</tr>

</table>

<center><h4>

<p align="center">NEW USER?<a href="Signup.html">REGISTER</a></p>

</form></h4>

</center>

<body>

</html>

**//update.html**

<html>

<head>

<title>Update page</title>

<script language="jscript">

var m,n;

function verifyItem()

{

m=document.myform.item.value;

if(m=="")

alert("enter item");

}

function verifyCost()

{

n=document.myform.cost.value;

if(n=="")

alert("enter cost");

}

</script>

</head>

<body style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<br></br><br></br>

<br></br><br></br>

<br></br><br></br>

<center><h1 style='font-family:"Courier New", Courier, monospace;'><caption>UPDATING THE ITEMS</caption></h1></center>

<br><br>

<form name="myform" method="post" action="update.com">

<center>

<table border="0" border="0" width="35%" height="15%" align="center"

style='font-family:"Courier New", Courier, monospace; font-size:150%'>

<tr>

<td>ITEM NAME:</td>

<td><input type="text" name="item" value="" size="30" >

</tr>

<tr>

<td>COST OF THE ITEM:</td>

<td><input type="text" name="cost" value="" size="30" OnClick="verifyItem()">

</tr>

<tr>

<td>AVAILABLE :</td>

<td><input type="text" name="available" value="" size="30"></td>

<tr></tr><tr></tr>

<tr></tr><tr></tr>

<tr></tr><tr></tr>

<tr>

<td rowspan="2">

<td><input type="submit" value="UPDATNG MENU" style="height:50px; width:150px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;"></td>

</tr>

</table>

</font>

</center>

</form>

</body>

</html>

**//Canteen.html**

<html>

<body text=white style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<form action="canteen.com" method="post">

<br></br>

<br></br>

<br></br>

<table align=center>

<td><input type="submit" value="GETMENU..!" style="height:50px; width:150px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;" ></td></tr>

</table>

</form>

</body>

</html>

**//Desc.html**

<html>

<head>

<style type="text/css">

img{

margin-right: 5em;

border:2px solid #000;}

</style>

</head>

<body style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<form method="get" action="login.html">

<center>

<br></br>

<h1><b style="font-size:30px;">

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbspCANTEEN AUTOMATION SYSTEM&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp

&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp</b>

<input type="submit" name="login" value="LOGIN" style="height:50px; width:75px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;" /></h1>

</center>

<br></br>

<center>

<h1><b style="font-size:30px;">DESCRIPTION</b></h1>

<br></br>

<p style = "font-family:georgia,garamond,serif;font-size:20px;font-style:italic;"><b>The Challenges encountered by the manual system in canteens is efficiency and customer satisfaction. The experience of ordering in canteens is not pleasant for customers as they have to wait in long queues before placing the order. And when the order is placed they have to wait near the counter until the order is prepared.</b> </p>

</center>

<br></br>

<marquee behavior="scroll" direction="left" loop="infinite" scrollamount="7" hspace=10 >

<img src="LIB/image/tiffin.jpg" width="120" height="80" alt="Natural"/>

<img src="LIB/image/sandwich.jpg" width="120" height="80" alt="Natural"/>

<img src="LIB/image/biryani.jpg" width="120" height="80" alt="Natural"/>

<img src="LIB/image/panipuri.jpg" width="120" height="80" alt="Natural" />

<img src="LIB/image/drinks.jpg" width="120" height="80" alt="Natural" />

<img src="LIB/image/frankie.jpg" width="120" height="80" alt="Natural" />

<img src="LIB/image/chocolate.jpg" width="120" height="80" alt="Natural" />

</marquee>

</body>

</html>

**//Food.html**

<html>

<body text=white style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<form action="food.com" method="post">

<br></br>

<br></br>

<br></br>

<table align=center>

<td><input type="submit" value="GETMENU..!" style="height:50px; width:150px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;" ></td></tr>

</table>

</form>

</body>

</html>

**//one.html**

<html>

<frameset rows="20%,\*">

<frame name="title" src="title.html">

<frame name="body" src="desc.html">

</frameset>

</html>

**//order.html**

<html>

<body text=white style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<form action="order.com" method="post">

<br></br>

<br></br>

<br></br>

<table align=center>

<td><input type="submit" value="GETORDERS..!" style="height:50px; width:150px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;" ></td></tr>

</table>

</form>

</body>

</html>

**//remove.html**

<html>

<head>

<title>remove page</title>

<script language="jscript">

var m,n;

function verifyItem()

{

m=document.myform.item.value;

if(m=="")

alert("enter item");

}

function verifyCost()

{

n=document.myform.cost.value;

if(n=="")

alert("enter cost");

}

</script>

</head>

<body style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<br></br>

<br></br>

<br></br>

<br></br>

<br></br>

<br></br>

<center><h1 style='font-family:"Courier New", Courier, monospace;'><caption>REMOVING THE ITEMS</caption></h1></center>

<br><br>

<form name="myform" action="remove.com" method="post">

<center>

<table border="0" width="35%" height="15%" align="center"

style='font-family:"Courier New", Courier, monospace; font-size:150%'>

<tr>

<td>ITEM NAME:</td>

<td><input type="text" name="item" value="" size="30">

</tr>

<tr>

<td>COST OF THE ITEM:</td>

<td><input type="text" name="cost" value="" size="30" OnClick="verifyItem()">

</tr>

<tr></tr>

<tr></tr>

<tr></tr>

<tr></tr>

<tr></tr>

<tr></tr>

<tr>

<td rowspan="2">

<td><input type="submit" value="REMOVE" style="height:50px; width:75px; font-size:10pt; color:black;

background-color:cyan; border:5px solid 336600; padding:7px;"> </td>

</tr>

</table>

</font>

</center>

</form>

</body>

</html>

**//Signup.html**

<html>

<head>

<script>

function VerifyDetails()

{

NAME=document.signupform.Name.value;

UID=document.signupform.uid.value;

EMAIL=document.signupform.email.value;

SIGNUPUSERNAME=document.signupform.signupusername.value;

SIGNUPPWD=document.signupform.signuppwd.value;

RE-SIGNUPPWD=document.signupform.re-signuppwd.value;

MOBILENO=document.signupform.mobileno.value;

if(NAME=="" || UID=="" || EMAIL==""|| SIGNUPUSERNAME=="" || SIGNUPPWD=="" || RE-SIGNUPPWD=="" || MOBILENO=="")

alert("enter all the details");

}

</script>

</head>

<body style="background-image:url(LIB/image/peach.png);

position:relative;

background-repeat:repeat;

background-position:center;">

<center><h1 style='font-family:

"Courier New",

Courier, monospace;'>

<caption>REGISTRATION FORM</caption></h1></center>

<center><br><br>

<form action="register.com" method=post>

<table border="0">

<tr>

<td>NAME :</td>

<td><input type="text" size="30" name="Name"></td>

</tr>

<tr>

<td>USER ID :</td>

<td><input type="text" size="30" name="uid"></td>

</tr>

<tr>

<td>BRANCH :</td>

<td>

<select name="branch">

<option value="none">NONE</option>

<option value="cse">CSE</option>

<option value="ece">ECE</option>

<option value="eee">EEE</option>

<option value="it">IT</option>

<option value="civil">CIVIL</option>

<option value="mech">MECH</option>

</select>

</td>

</tr>

<!--<tr>

<td>Section</td>

<td>

<select>

<option value="none">NONE</option>

<option value="a">A</option>

<option value="b">B</option>

<option value="c">C</option>

<option value="d">D</option>

<option value="e">E</option>

<option value="f">F</option>

</select>

</td>

<td>If faculty(select department)</td>

<td>

<select>

<option value="none">NONE</option>

<option value="cse">CSE</option>

<option value="ece">ECE</option>

<option value="eee">EEE</option>

<option value="it">IT</option>

<option value="civil">CIVIL</option>

<option value="mech">MECH</option>

</select>

</td>

</tr>-->

<tr>

<td>EMAIL</td>

<td><input type="text" size="30" name="email"></td>

</tr>

<tr>

<td>USERNAME : </td>

<td><input type="text" size="30" name="signupusername"></td>

</tr>

<tr>

<td>PASSWORD :</td>

<td><input type="password" size="30" name="signuppwd"></td>

<tr>

<td>RE-ENTER THE PASSWORD :</td>

<td><input type="password" size="30" name="re-signuppwd"></td>

</tr>

<tr>

<td>MOBILE NUMBER:</td>

<td><input type="text" size="30" name="mobileno"></td>

</tr>

<tr>

<td></td>

<td><input type="submit" name="register" value="REGISTER" style="height:50px; width:150px; font-size:10pt; color:black;

background-color:cyan;

border:5px solid 336600;

padding:7px;"

OnClick="VerifyDetails()"></td>

</tr>

</table>

</form>

</center>

</html>

**//title.html**

<html>

<body background="LIB/image/white.png">

<table border="0" cellspacing="0"

width="100%">

<tr>

<td><img src="LIB/image/white.png" width="250px" height="125px">

<td><img src="LIB/image/grietcse.png" width="1000px" height="125px">

<td><img src="LIB/image/white.png" width="250px" height="125px">

</tr>

</table>

</body>

</html>

**//Pinsert.sql**

Create or replace procedure Pinsert(pserialnumber in details\_db.serialnumber%type, puserid in details\_db.userid\_ldb%type, pname in details\_db.name\_ddb%type, pbranch in details\_db.branch\_ddb%type, pemail in details\_db.email\_ddb%type, pphonenumber in details\_db.phonenumber\_ddb%type, pusername in details\_db.username\_ldb%type)

IS

Begin

Insert into details\_db values(pserialnumber,puserid,pname,pbranch,pemail,pphonenumber,pusername);

End;

/

**//ldelete.sql**

create or replace procedure ldelete(ptokenid in order\_db.token%type)

IS

Begin

delete from order\_db where token=ptokenid;

End;

/

**//Pdelete.sql**

create or replace procedure Pdelete(pitemname in food\_tdb.itemname\_fdb%type)

IS

Begin

delete from food\_tdb where itemname\_fdb=pitemname;

End;

/

**//Pinsertldb**

create or replace procedure Pinsertldb(pusername in login\_db.username\_ldb%type,ppassword in login\_db.password\_ldb%type)

IS

Begin

insert into login\_db values (pusername,ppassword);

End;

/

**//POinsert.sql**

create or replace procedure POinsert(ptoken in order\_db.token%type,pitem1 in order\_db.item1%type,pitem2 in order\_db.item2%type,pitem3 in order\_db.item3%type,pitem4 in order\_db.item4%type,pitem5 in order\_db.item5%type,pcost in order\_db.cost%type)

IS

Begin

insert into order\_db values(ptoken,pitem1,pitem2,pitem3,pitem4,pitem5,pcost);

End;

/

**//Pupdate**

create or replace procedure PUpdate(pitemname in food\_tdb.itemname\_fdb%type,pavailable in food\_tdb.availability\_ftdb%type)

IS

Begin

update food\_tdb set availability\_ftdb=pavailable where itemname\_fdb=pitemname;

End;

/

**//web.xml**

<web-app>

<welcome-file-list>

<welcome-file>one.html</welcome-file>

</welcome-file-list>

<servlet>

<servlet-name>Login</servlet-name>

<servlet-class>LoginValidation</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Login</servlet-name>

<url-pattern>/login.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>Signup</servlet-name>

<servlet-class>SignupSql</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Signup</servlet-name>

<url-pattern>/register.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>food</servlet-name>

<servlet-class>FoodServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>food</servlet-name>

<url-pattern>/food.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>next</servlet-name>

<servlet-class>NextServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>next</servlet-name>

<url-pattern>/next.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>canteen</servlet-name>

<servlet-class>CanteenServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>canteen</servlet-name>

<url-pattern>/canteen.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>remove</servlet-name>

<servlet-class>RemoveServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>remove</servlet-name>

<url-pattern>/remove.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>update</servlet-name>

<servlet-class>UpdateServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>update</servlet-name>

<url-pattern>/update.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>Order</servlet-name>

<servlet-class>OrderServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Order</servlet-name>

<url-pattern>/orders.com</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>Done</servlet-name>

<servlet-class>DoneServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Done</servlet-name>

<url-pattern>/lastdelete.com</url-pattern>

</servlet-mapping>

</web-app>

**6. TESTING**

**6.1 SYSTEM TESTING:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**6.2 TYPES OF TESTS:**

**Unit Testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration Testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional Test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Testing:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach:**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives:**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested:**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

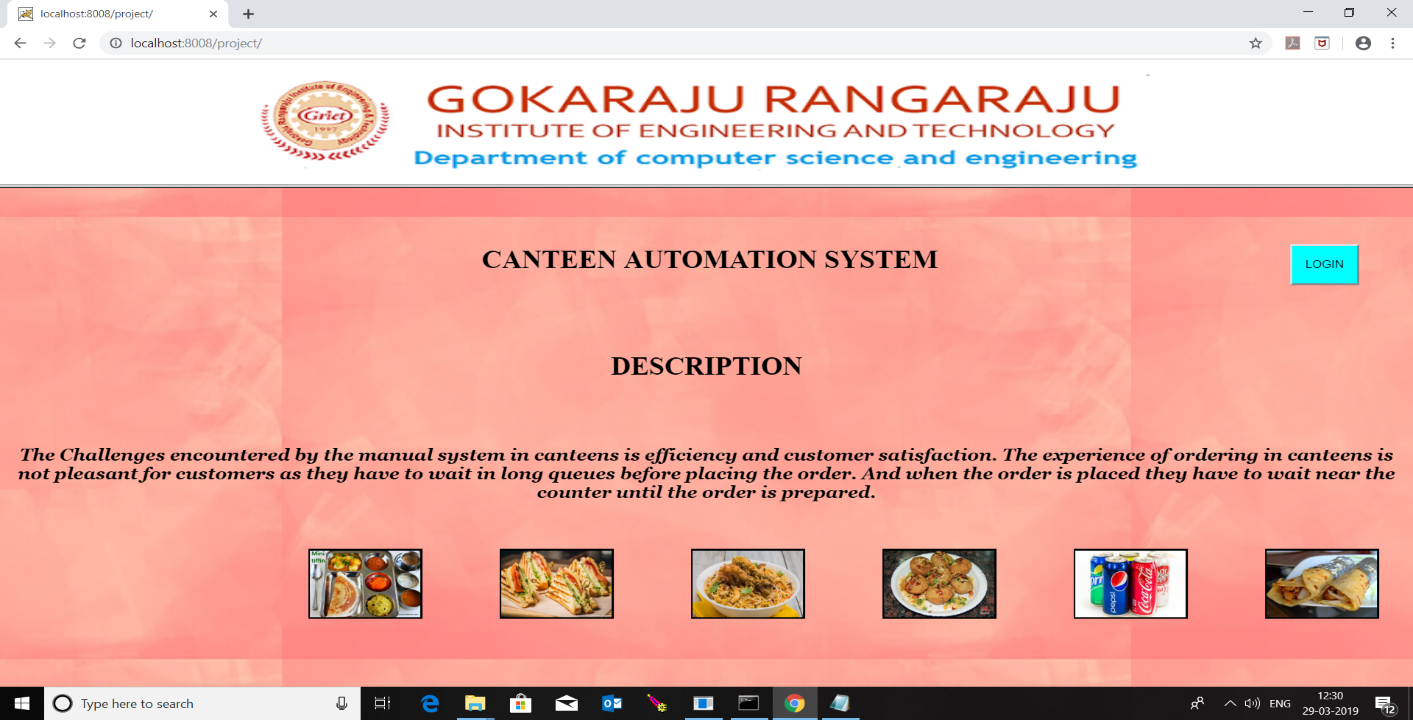
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

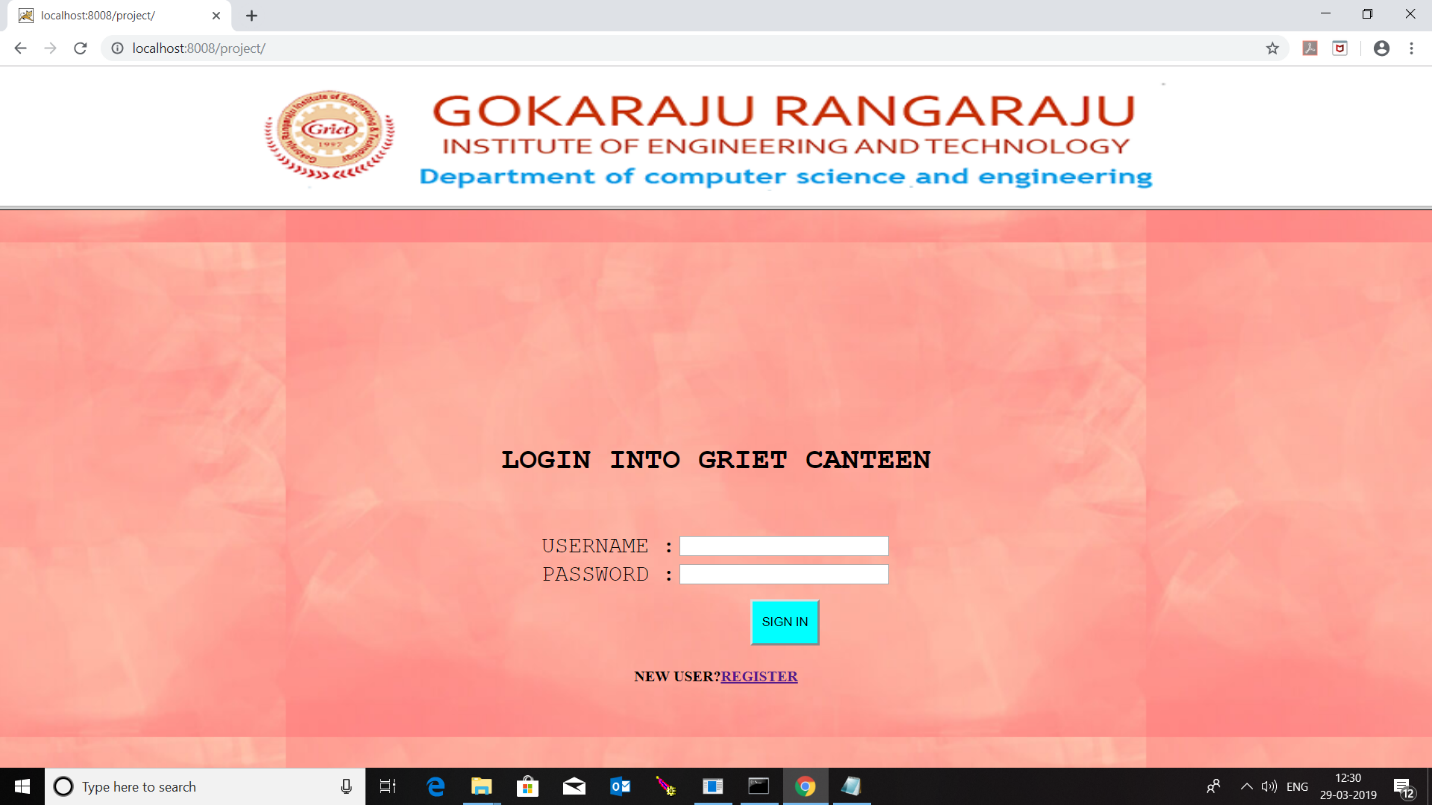
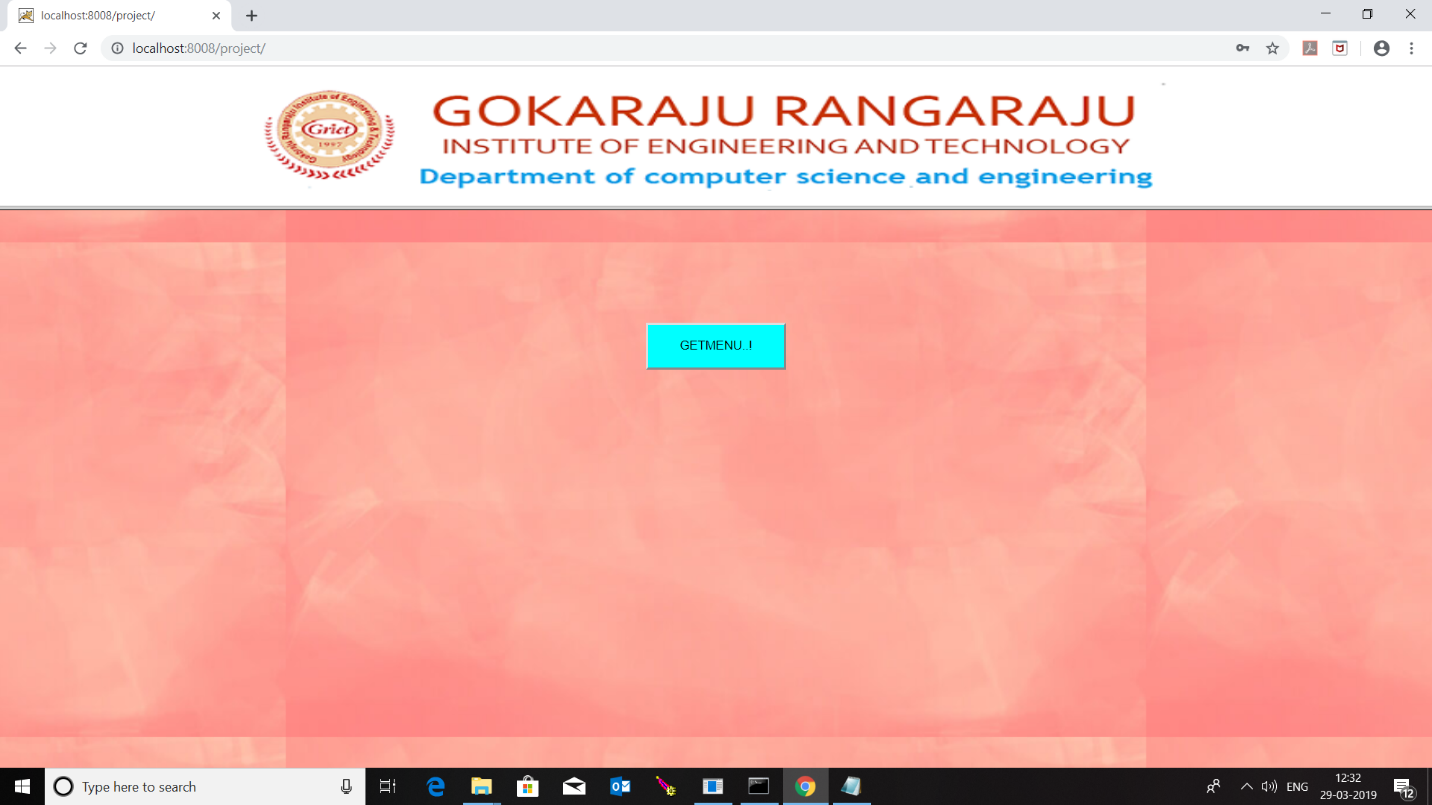
**Acceptance Testing:**

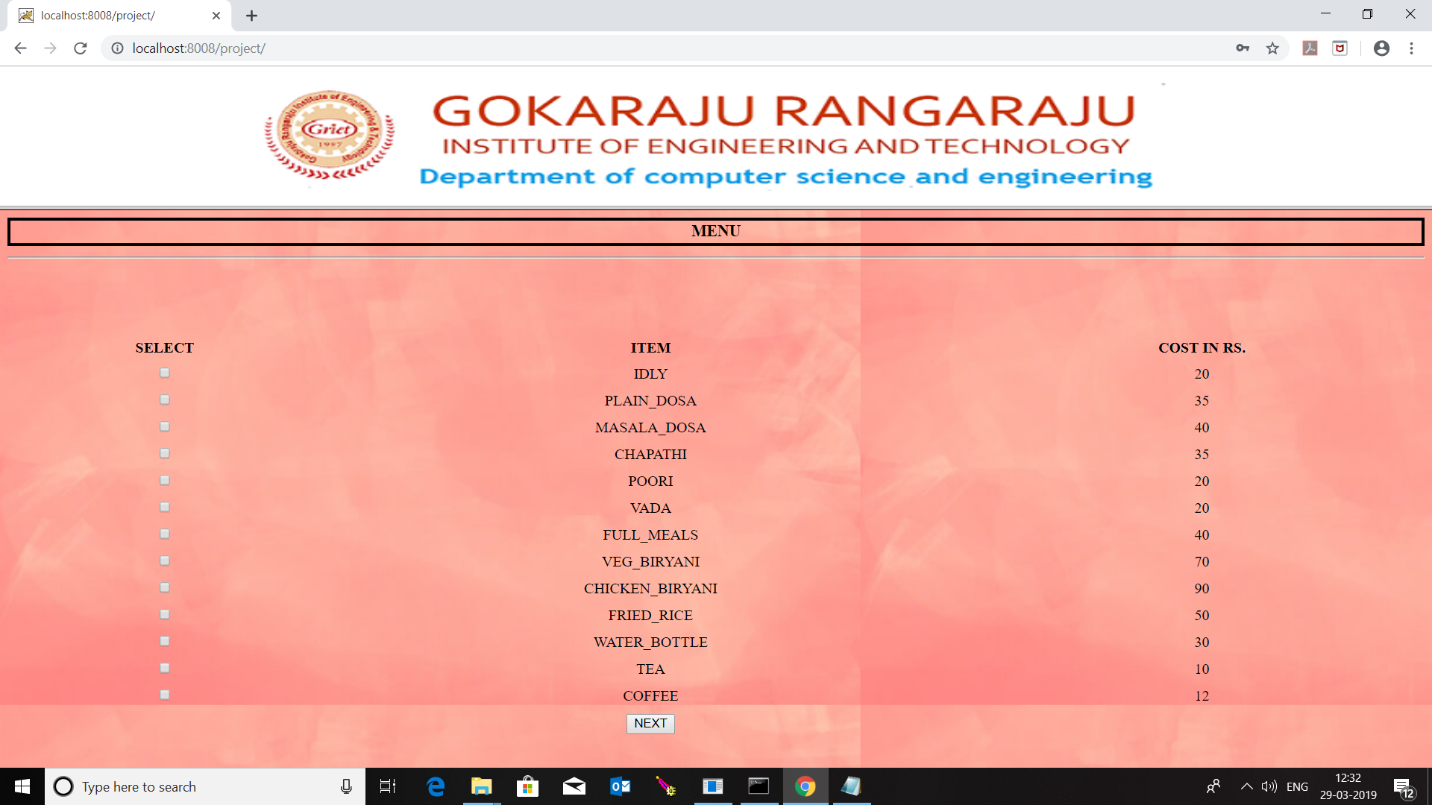
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

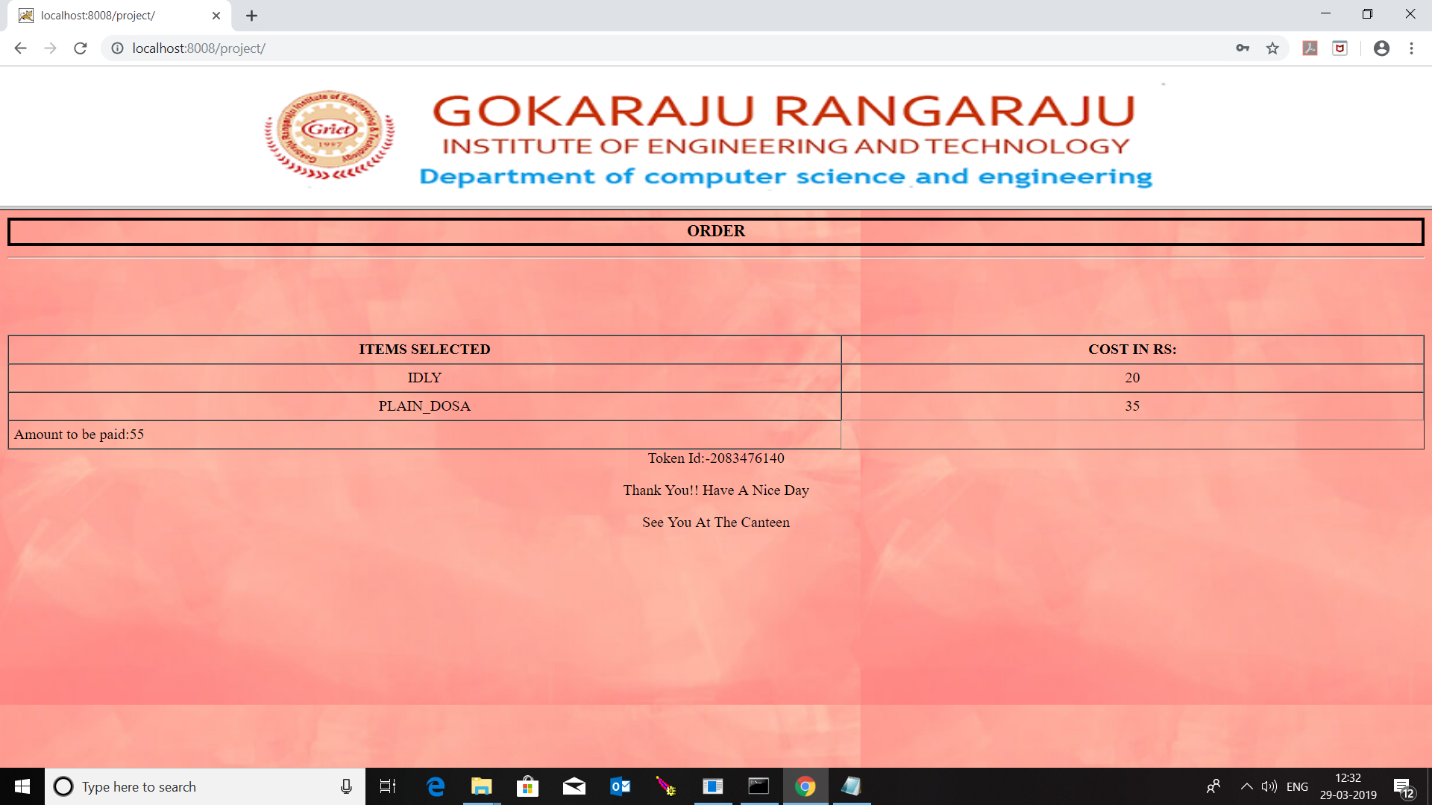
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7. SCREENSHOTS**

**Fig – 7.1 Home page**

**Fig – 7.2 Login PageFig – 7.3 Registration PageFig – 7.4 Get Menu Page**

****

**Fig – 7.5 Menu List Page Fig – 7.6 Token Page Fig – 7.7 Canteen End Page Fig – 7.8 Show Orders Page**

****

**Fig – 7.9 Update Items Page**

****

**Fig – 7.10 Remove Items Page**

**8. CONCLUSION**

**Time Saving:** Our canteen automation system is developed with a primary aim of 'Saving Time. The customer can order the food and it is also efficient for canteen workers because this system takes lesser time as compared to phone based or manually based system. No Complication: Major complication part for canteen automation system is adding a product or managing products section. In our canteen automation system, no complicated part is involved in managing sections; we have taken immense care in this section and nullified all complications

which make this system accurate and unique.

**Cost Effective:** It's cheaper. You don't have to purchase multiple copies of software to install on multiple computers. Multiple copies often require you to pay multiple licensing fees, but since you aren't actually purchasing any software with an online system, that's not a concern.

**Security:** Online systems are just as secure. Most online canteen automation system programs allow you to create multiple user accounts with various levels of access. Your data is stored on secure, protected servers that feature firewalls and other online security programs.

**9. BIBLIOGRAPHY**

* Robin Nixon(2010). MySQL, JavaScript &CSS
* Ian Sommerville(2010).SoftwareEnginering
* en.wikipedia.org
* Java The complete reference-7th edition by Herbert Schildt