ENTERPRISE RESOURCE PLANNING FOR DAIRY

A PROJECT REPORT

Submitted by

SRI ANUSHA V	71205104048
SUBIKSHA M	71205104054

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2009

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**ENTERPRISE RESOURCE PLANNING FOR DAIRY**" is the bonafide work of "**SRI ANUSHA V** and **SUBIKSHA M**" who carried out the project under my supervision.

SIGNATURE

Dr. S.Thangasamy

DEAN

Department of Computer Science & Engg
Kumaraguru College of Technology,
Chinnavedampatti Post,
Coimbatore – 641 606

SIGNATURE

Mrs. R.Kalaiselvi

SUPERVISOR

Senior Lecturer

Department of Computer Science & Engg
Kumaraguru College of Technology,
Chinnavedampatti post,
Coimbatore – 641 606

The candidates with University Register Nos. **71205104048** and **71205104054** were examined by us in the project viva-voce examination held on ...27.04.2007....

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be but incomplete without the mention of the people who made this possible and whose constant guidance and encouragement crowns all efforts with success.

We are extremely grateful to **Dr.Annamalai,** Vice Principal, Kumaraguru College of Technology for having given us this opportunity to embark on this project.

We express our sincere and heartfelt thanks to **Dr.S.Thangaswamy**, Dean, Department of Computer Science and Engineering, for his kind guidance and support.

We would like to express our sincere thanks to our project coordinator **Mrs.P.Devaki**, for her valuable guidance during the course of the project.

We would also like to thank our class advisor **Mrs.D.Chandrakala,** for her constant support and guidance.

We would like to thank our guide **Mrs.R.Kalaiselvi**, without whose motivation and guidance we would not have been able to embark on a project of this magnitude. We express our sincere thanks for her valuable guidance, benevolent attitude and constant encouragement.

We reciprocate the kindness shown to us by the staff members of our college, people at home and our beloved friends who have contributed in the form of ideas, constructive criticisms and encouragements for the successful completion of the project.

ABSTRACT

ABSTRACT

The project includes the management of various processes involved in a milk industry. Processes like obtaining the milk from the farmer, testing the milk for quality parameters and controlling the machinery are managed with the help of computers in this project. The quality parameters of milk from each milch animal varies everyday depending upon various parameters such as genetic potential of animal, feeding habits, interval between milking, etc., Hence the milk has to be tested at the farmer level for such quality parameters. The pricing is done with the help of these parameters.

The milk reaches the dairy, collection center-wise. It is weighed and tested for various quality parameters and is accordingly taken for further processing. The processing includes chilling, pasteurization, clarification, homogenization and packing to various sizes in packing machines. The automation of testing machinery and the controlling of the milk flow are carried out with the help of computers.

LIST OF TABLES

LIST OF ABBREVIATIONS

LIST OF ABBREVIATIONS

SNF	-	Solids Non Fat
kgTS	-	Kilogram of Total Solids
CLR	-	Corrected Lactometer Reading

CONTENTS

TABLE OF CONTENTS

INTRODUCTION

INTRODUCTION

Milk is a universal food. It is a complete food for all age groups. Milk and milk products represent the most prolific segment of our Indian Dairy Industry. Dairying has played a prominent role in strengthening our rural economy. Milk is one of the commodities that is perishable hence it has to be processed and delivered to the consumer in time.

Milk is one of the perishable commodities. It has to be brought to the dairy for processing within a time span of 4 to 5 hours from the time of milking. The general processes involved in the processing of milk include

- Chilling
- Filtration
- Clarification
- Separation
- Standardization
- Homogenization
- Pasteurization
- Packing

CHILLING

The raw milk received at the processing unit (Dairy) is immediately chilled to 4 degree Celsius in order to prolong the shelf life of milk.

FILTRATION

The purpose of filtration is to remove visible particles and dirt that may have found access to milk. Filtration is usually carried out at 32 to 42 degree Celsius. Higher temperatures are avoided because of possible damage to the creaming property of the milk.

CLARIFICATION

Filtration removes sediment but does not remove the leucocytes (under tissue). They are removed by "clarification". Clarification also removes other large cells and fine dirt.

SEPARATION

The process of separating Fat from milk is known as separation. It is possible because of a difference in specific gravity between the Fat and the liquid portion of the milk.

STANDARDIZATION

A dairy plant receives milk from various dairy farms having varying Fat and SNF (Solids Non Fat) content. Thus the Fat content of the mixed lot varies from day to day. This may cause loss to either the plant or the consumer, if the milk is not standardized to a particular Fat content.

HOMOGENIZATION

In homogenization, the Fat globules of milk are broken down to smaller sizes, preferably below 2 microns, so that they no longer exhibit a tendency to cluster or rise.

PASTEURIZATION

Milk produced from normal, healthy milch animals generally does not contain pathogens, these invariably get access to the milk through various means while handling. All these pathogens are destroyed by heat treatment. Pasteurization is designated by temperature-time combination and also includes cooling of milk to the safe temperature.

PACKING

The chilled milk is either stored in storage tanks or sent to packing section. After packing, the pouches are put in crates and are kept in the cold-store. The crates are then later distributed to the dealers in the market.

LITERATURE REVIEW

LITERATURE REVIEW

PROBLEM DOMAIN

The dairy industry consists of many machineries and equipments which are to be monitored on a continuous basis. The tedious routine leads to the entering of incorrect information into the database.

The milk is collected by the various collection centers. In each collection center, the quantity of milk poured by each farmer and the milk's quality parameters such as Fat and SNF (Solids Non Fat) are penned down. The lactometer reading gives the SNF content. The quantity of milk poured is measured using litre jars. The milk is then brought to the dairy for processing.

The milk at the reception dock is tested for quality parameters. The values of these parameters are cross checked with those obtained from the collection centers. The pricing for each farmer and each center is done based on these parameters. These operations are performed manually and are entered into the system. There are possibilities of confusion in the script, manipulation of data and entering of incorrect data into the system.

The milk flow is controlled manually by turning the valves. Our project determines to create a web based application for entering the information, storing the data, viewing the details and controlling the flow of milk using computers.

EXISTING SYSTEM

In the existing system, the values of quality parameters of milk such as Fat and SNF and the amount of milk poured by each farmer are penned down at each collection center. The milk that reaches the processing unit is also tested for quality parameters and compared with the values obtained from the collection centers. These operations are performed manually. Later these details are fed into the database for future reference.

At the milk reception dock, milk is poured into the weigh bowl collection center-wise. The quantity of milk present is noted down and a sample is taken to find out the Fat and SNF values. The milk is then processed. The control of milk flow in the existing system is performed manually by turning the valves.

Problem Faced:

To err is human. Mistakes made while penning down of data are unavoidable which might result in huge loss. Controlling of values and maintaining of temperature should be done effectively. Delay in any of the dairy process leads to drastic changes in quality parameters.

PROPOSED SYSTEM

Our project determines to create a web based application for entering the information, storing the data, viewing the details and controlling the flow of milk in a dairy plant using computers.

Each collection center is provided with a computer with internet facility. The information fed in each collection center is stored in the central database. The quantity of milk poured into the weigh bowl at the milk reception dock is directly captured by the computer. Similarly the Fat and SNF values of milk are also captured. These are made possible by connecting the weigh bowl and the milk testers with the computer.

Thus the quality parameters of milk are obtained. The pricing is now done with the help of a programming module. The flow of milk is controlled by switching on and switching off the pumps connected to each machine. The switching on and switching off of the pumps are carried out with the help of computers.

Advantages:

- Minimized work force
- Information directly fed to database
- Accurate data
- No misinterpretation of data
- Fast and efficient operation

SCOPE OF THE PROJECT

Today's world is looking out for better living style. Things produced should be in a faster phase for the growing population. Our system aims in fulfilling this need by improving the dairy processes in an efficient manner. The operations performed by computer are faster and more accurate. The controlling of machineries with the help of computer is more effective. Thus it results in the milk being processed and delivered to the consumer in a better manner.

SYSTEM ANALYSIS

SYSTEM ANALYSIS

PROJECT DESCRIPTION

The project work entitled "ENTERPRISE RESOURCE PLANNING FOR DAIRY" is a web based application. This project deals with producers, manufactures and the machineries. It includes entering, deleting and viewing of centre information, farmer information and procurement details. It also involves comparison of data obtained from collection center with the data obtained from the dairy. In this project, the flow of milk is controlled with the help of computers.

The different modules present in the project are:

- **Home**

  This shall be the front page that provides information about the different pages in the website and has hyperlinks to all the pages.

- **Center Information**

  This link contains information about the collection centers. The details of the collection center namely the center id, name of the center and its location are obtained using this link. It is possible for the administrator to add a new center, edit the existing center information and remove a particular center from the database.

- **Farmer Information**

This link contains details about the farmer namely the farmer id, name of the center where he pours milk, farmer address and the date of joining of the farmer are obtained using this link.

- **Procurement Details**

The procurement link is used to enter the procurement details, view the details of farmer and center by date. The value of rate per kgTS can be changed in this link. It displays the values of rate per litre and the total amount for the quantity of milk poured by a farmer.

- **Compare**

This link is used to compare the quality parameters of milk for a particular center. Here the values obtained from the collection center are compared with those obtained from the dairy. The values are checked to verify whether any adulteration has taken place during the transportation of milk from collection center to the dairy.

- **Processing**

This link deals with the automation of machines with the help of computers. The values of quality parameters of milk are directly stored in the database. It includes the controlling of milk flow by using computers.

SOFTWARE DESCRIPTION

SOFTWARE DESCRIPTION

APACHE TOMCAT

Programs need an environment to run within a host computer. Sometimes the operating system is sufficient to provide the environment, but at other times, a more sophisticated container is needed. Tomcat is a container that is used to provide an environment for Java code running on a web server.

Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and Javaserver Pages specifications are developed by Sun under the Java Community Process.

Using Port 8080:

On a Windows machine the port attribute of the connector within the *Catalina* service from 8080 is available. This allows usage of tomcat directly to serve all requests.

Features of Tomcat 5.x :

- Implements the Servlet 2.4 and JSP 2.0 specifications
- Reduced garbage collection, improved performance and scalability
- Native Windows and Unix wrappers for platform integration
- Faster JSP parsing

JAVA

Java first became popular by being the earliest portable dynamic client-side content for the World-Wide Web in the form of platform-independent Java "applets". In the late 1990's and into the 2000's it has also become very popular on the server side, where an entire set of APIs defines the J2EE.

- Java is an object-oriented programming language developed by Sun Microsystems
- Java was modeled after C++
- Java is designed to be small, simple, and portable across platforms and operating systems
- Java is used to develop executable, distributed applications for delivery to a Java-enabled Web browser or the Java Interpreter. A Java programmer can create the following:
    - **applets:** Programs that are called through an HTML page and run on a Java-enabled browser.
    - **applications:** Standalone Java programs executed independently of a browser. The execution is done using the Java interpreter.
- Java brings interactivity into the Web

There are also packages for developing XML applications, web services, servlets and other web applications, security, date and time calculations and I/O formatting, database (JDBC), and many others.

FEATURES OF JAVA

- **Simple**
    - Java is a safer and cleaner language than C or C++.
    - Memory is managed automatically so the programmer doesn't have to worry about freeing the unused space.

- **Object Oriented**
    - Object-oriented programming is based upon modeling the world in terms of software components called objects.
    - An object consists of data and methods.
    - Methods are operations that can be performed on that data.

- **Distributed**
    - Java is specifically designed to work within a network environment.
    - Java has a large library of classes to handle TCI/IP, HTTP, FTP and other networking protocols.

- **Secure**
    - Java was designed with the knowlegde that the applications will be transferred through the network.
    - Points of entry to protected sectors of memory used by viruses and Trojan horses are impossible to reach using Java.

- **Multithreaded**
  - Multithreading is the ability for one program to do more than one thing at once.
  - Threads are easy to manage in Java and they take advantage of multiprocessor systems if the operating system does so.
  - Threads in java bring better interactive responsiveness and real-time behavior.

- **Portable**
- **Dynamic**
  - Java was designed to adapt to an evolving environment.
  - If you make changes to a parent class in most instances it will not affect the already existing applications. A change of this magnitude in C++ will normally involve recompiling the whole application.
  - In Java you can add new methods and instance variables to libraries without affecting the client applications.

- **Robust**
  - Java unlike C or C++ (in some instances) is more careful at handling data types.
  - Java does not support pointers, it uses arrays instead.
  - Java won't allow overwriting of memory and corrupting of data through pointers.

ORACLE

The Oracle Database consists of a relational database management system (RDBMS) produced and marketed by Oracle Corporation.

The Oracle RDBMS stores data logically in the form of tablespaces and physically in the form of data files. Tablespaces can contain various types of memory segments, such as Data Segments, Index Segments, etc. Segments in turn comprise one or more extents. Extents comprise groups of contiguous data blocks. Data blocks form the basic units of data storage. At the physical level, datafiles comprise one or more data blocks, where the block size can vary between data-files.

Oracle database management tracks its computer data storage with the help of information stored in the SYSTEM tablespace. The SYSTEM tablespace contains the data dictionary and often indexes and clusters. A data dictionary consists of a special collection of tables that contains information about all user-objects in the database.

In version 10g, grid computing has introduced shared resources where an instance can use (for example) CPU resources from another node (computer) in the grid.

The Oracle DBMS can store and execute stored procedures and functions within itself. PL/SQL (Oracle Corporation's proprietary procedural extension to SQL), or the object-oriented language Java can invoke such code objects and/or provide the programming structures for writing them.

Oracle instance frequently accesses the data dictionary in order to parse SQL statements. The operation of Oracle depends on ready access to the data dictionary: performance bottlenecks in the data dictionary affect all Oracle users. Because of this, database administrators should make sure that the data dictionary cache has sufficient capacity to cache this data. Without enough memory for the data-dictionary cache, users see a severe performance degradation. Allocating sufficient memory to the shared pool where the data dictionary cache resides precludes these particular performance problems.

Apart from the clearly-defined database options, Oracle databases may include many semi-autonomous software sub-systems, which Oracle Corporation sometimes refers to as "features" in a sense subtly different from the normal usage of the word. Such "features" may include (for example):

- Automatic Workload Repository (AWR), providing monitoring services to Oracle database installations from Oracle version 10. Prior to the release of Oracle version 10, the Statspack facility provided similar functionality.

- Cluster ware
- Data Aggregation and Consolidation
- Data Guard for high availability
- Generic Connectivity for connecting to non-Oracle systems.
- Data Pump utilities, which aid in importing and exporting data and metadata between databases
- Database Resource Manager (DRM), which controls the use of computing resources.
- Fine-grained auditing (FGA) (in Oracle Enterprise Edition) supplements standard security-auditing features
- Flashback for selective data recovery and reconstruction
- iSQL*Plus, a web-browser-based graphical user interface (GUI) for Oracle database data-manipulation (compare SQL*Plus)
- Oracle-managed files (OMF) -- a feature allowing automated naming, creation and deletion of data files at the operating-system level.
- Recovery Manager (rman) for database backup, restoration and recovery

DEVELOPMENT PHASE

DEVELOPMENT PHASE

TABLE DESCRIPTION

CENTER

Field Name	Field Type	Comments
center_id	number	center’s id
centre_name	text	center’s name
center_loc	text	center’s location

FARMER

Field Name	Field Type	Comments
center_id	number	center’s id
farmer_id	number	farmer’s id
farmer_name	text	farmer’s name
farmer_addr	text	farmer’s address

RATEKGTS

Field Name	Field Type	Comments
value	number	rate per kgTS

PROCURE

Field Name	Field Type	Comments
center_id	number	center's id
farmer_id	number	farmer's id
quan	number	quantity of milk
clr	number	milk's CLR
fat	number	Fat content in milk
snf	number	SNF content in milk
lrate	number	rate per litre
totrate	number	total amount for the total quantity of milk
dt	number	date of pouring milk

COMPARE

Field Name	Field Type	Comments
center_id	number	center's id
quan	number	quantity of milk
fat	number	Fat content in milk
snf	number	SNF content in milk
dt	number	date of pouring milk

COMPARE1

Field Name	Field Type	Comments
center_id1	number	center's id
quan1	number	quantity of milk
fat1	number	Fat content in milk
snf1	number	SNF content in milk
dt 1	number	date of pouring milk

RESULT

Field Name	Field Type	Comments
center_id	number	center's id

SYSTEM TESTING

SYSTEM TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Thus the system test in implementation should be a confirmation that everything is correct. It is an opportunity to show the user that the system works.

We have tested each page individually using the test data designed by us. The complete system and its environment were tested to the satisfaction of the system analyst and the user. It was shown to the system analyst that the system would operate successfully in all aspects and produce expected results under expected conditions.

SCOPE FOR FUTURE ENHANCEMENTS

SCOPE FOR FUTURE ENHANCEMENTS

The project has been completed successfully and all the requirements have been met. However the possibilities for renovation are infinite and the scope for development is innumerable.

The transportation module which includes a program for the distribution of milk to dealers in various markets can be created. The routes are formed based on logistics using transportation module. The dealer-wise, product-wise daily off-take, payment collection, crates movement, leakage, discount, promotional schemes can be computerized and monitored.

CONCLUSION

CONCLUSION

Our attempt to create a software for the automation of the dairy has been successful and we take great pride in presenting to you “ENTERPRISE RESOURCE PLANNING FOR DAIRY”. This system is created to work in real time environment.

A proper software development scheme is employed in developing the application and the system is tested for various inputs. All the exceptions that occurred during the implementation are handled as required.

The performance of the devised system is compared with the existing system and its efficiency is found to be much better.

APPENDIX

APPENDIX

SAMPLE CODE

homepage.html

```
<html>
<frameset cols="25%,75%">
<frame noresize="noresize" src="main.html">
<frame noresize="noresize" src="frame_b.html"name="showframe">
<body>
</body>
</frameset>
</html>
```

main.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink="
#339999" alink="#996699">
<br>
<p align="center">
<font face="Monotype corsiva"><font color="#009999"><font size
="7">MY DAIRY</font>
</p>
<br><br>
<p>
<font face="monotype corsiva"><font size="5">
```

```
<a href ="center.html" target ="showframe">Center Information </a>
<br><br>
<a href ="farmer.html" target ="showframe">Farmer  Information </a>
<br><br>
<a href ="procure.html" target ="showframe"> Procurement </a> <br> <br>
<a href ="process.html" target ="showframe">Processing</a><br><br>
<a href ="compare.html" target ="showframe">Compare</a><br><br>
<a href ="result.html" target ="showframe">Result</a><br><br>
<a href ="connectivity.html" target ="showframe">Connectivity </a>
</font></p>
</body>
</html>
```

frame_b.html

```
<html>
<body background="15720.png">
</body>
</html>
```

center.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink =
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
```

```
<font face="Monotype corsiva"><font color="#009999"><font
size="5">Center Info</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<a href ="center_view.html" target ="showframe"> View </a> <br><br>
<a href ="center_add.html" target ="showframe">Add</a><br> <br>
<a href ="center_remove.html" target ="showframe">Remove</a>
</font>
</pre>
</p>
</body>
</html>
```

farmer.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
<font face="Monotype corsiva"><font color="#009999"><font
size="5">Farmer Info</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<a href ="farmer_view.html" target ="showframe">View</a><br> <br>
```

```
<a href ="farmer_add.html" target ="showframe">Add</a><br> <br>
<a href ="farmer_remove.html" target ="showframe">Remove</a>
</pre>
</p>
</body>
</html>
```

procure.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
<font face="Monotype corsiva"><font color="#009999"><font
size="5">Procurement Details</font>
</p><p>
<font face="monotype corsiva"><font size="5">
<a href ="procure_enter_details.html" target ="showframe">Enter
Details</a><br><br>
<a href ="view_details.html" target ="showframe">View Details
</a><br><br>
<a href ="Rate.html" target ="showframe">Change Rate</a>
</pre>
</p>
</body>
</html>
```

process.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
<font face="Monotype corsiva"><font color="#009999"><font
size="5">Processing</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="newprocess">
Enter password     <input type="password" name="id">
                   <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

compare.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
```

```
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Compare</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="compare">
        Enter Center Id     <input type="text" name="center_id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

result.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Result</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
```

```
<form method="get"action="newresult">
        Enter password     <input type="password" name="id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

connectivity.html

```
<html>
<head>
<script>
function f1()
{
        alert("Switch ON Collection Bin");
        document.form1.button.value="button1";


}
function f2()
{
        alert("Switch OFF Collection Bin");
        document.form1.button.value="button2";
}
function f3()
{
        alert("Switch ON Dump Tank");
        document.form1.button.value="button3";
}
function f4()
{
        alert("Switch OFF Dump Tank");
        document.form1.button.value="button4";
}
function f5()
{
        alert("Switch ON Silo1");
        document.form1.button.value="button5";
}
function f6()
{
        alert("Switch OFF Silo1");
        document.form1.button.value="button6";
}
function f7()
{
        alert("Switch ON Silo2");
        document.form1.button.value="button7";
}
function f8()
{
        alert("Switch OFF Silo2");
        document.form1.button.value="button8";
}
```

```
</script>
</head>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<form method="get"action="connect" name=form1>
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Controls</font>
</p>
<p><font face="monotype corsiva"><font size="5"><input type= "hidden"
name="button">

Collection bin      <input type="submit" value="ON" onClick="f1()">
                    <input type="submit" value="OFF" onClick="f2()">
Dump Tank           <input type="submit" value="ON" onClick="f3()">
                    <input type="submit" value="OFF" onClick="f4()">
Silo1               <input type="submit" value="ON" onClick="f5()">
                    <input type="submit" value="OFF" onClick="f6()">
Silo2               <input type="submit" value="ON" onClick="f7()">
                    <input type="submit" value="OFF" onClick="f8()">
</pre>
</p>
</form>
</body>
</html>
```

center_add.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Add New Center</font>
</p>
<p><font face="monotype corsiva"><font size="5">
<form method="get"action="centeradd">
        Center Id               <input type="text" name="center_id">
        Center Name             <input type="text" name= "center_name">
        Center Location         <input type="text" name="center_loc">
<input type="submit" value="OK">   <input type="reset" value="Clear">
</pre>
</p>
</form>
</body>
</html>
```

center_remove.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
```

```
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Remove Center</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="centerremove">
        Center Id    <input type="text" name="center_id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

center_view.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">View Center Info</font>
</p>
<p>
```

```
<font face="monotype corsiva"><font size="5">
<form method="get"action="centerview">
        Center Id    <input type="text" name="center_id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

farmer_add.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Add New Farmer</font>
</p><p><font face="monotype corsiva"><font size="5">
<form method="get"action="farmeradd">
        Center Id               <input type="text" name="center_id">
        Farmer Id               <input type="text" name="farmer_id">
        Farmer Name             <input type="text" name= "farmer_name">
        Farmer Address          <input type="text" name="farmer_addr">
<input type="submit" value="OK">  <input type="reset" value="Clear">
</pre>
```

```
</p>
</form>
</body>
</html>
```

farmer_remove.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699"
vlink="#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Remove Farmer</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="farmerremove">
        Farmer Id    <input type="text" name="farmer_id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

farmer_view.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">View Farmer Info</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="farmerview">
        Farmer Id    <input type="text" name="farmer_id">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

procure_enter_details.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<p>
```

```
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Enter Details</font>
</p><p><font face="monotype corsiva"><font size="5">
<form method="get"action="procuredetails">
        Center Id           <input type="text" name="center_id">
        Farmer Id           <input type="text" name="farmer_id">
        Quantity            <input type="text" name="quan">
        CLR                 <input type="text" name="clr">
        Fat                 <input type="text" name="fat">
<input type="submit" value="OK">  <input type="reset" value="Clear">
</pre>
</p>
</form>
</body>
</html>
```

rate.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<form method="get"action="rate">
<pre>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Enter Rate per KgTS</font>
</p><p><font face="monotype corsiva"><font size="5">
<form method="get"action="procuredetails">
```

```
        Rate per KgTS             <input type="text" name="rateval">
        <input type="submit" value="OK">   <input               type="reset"
value="Clear"></pre>
</p>
</form>
</body>
</html>
```

output.html

```
<html>
<body>
<form>
<pre>
<p>
<font face="Monotype corsiva"><font color="#009999"><font size="5">
Center Id           <input type="text" name="centre_no">
Center Name               <input type="text" name="farmer_id">
Center Location     <input type="text" name="quantity">
</font></p>
</pre>
</form>
</body>
</html>
```

view_bydate_bycenter.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
<br><br>
<p>
        <font face="Monotype corsiva"><font color="#009999"><font
size="5">Procurement Details of Center by date</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="viewbycenter">
        From Date               <input type="text" name="fromdate">
        To Date                 <input type="text" name="todate">
        <input type="submit" value="OK">
</pre>
</p>
</form>
</body>
</html>
```

view_bydate_byfarmer.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre>
```

```
<br><br>
<p><font face="Monotype corsiva"><font color="#009999"><font
size="5">Procurement Details of Farmer by date</font>
</p>
<p>
<font face="monotype corsiva"><font size="5">
<form method="get"action="viewbyfarmer">
        From Date        <input type="text" name="fromdate">
        To Date          <input type="text" name="todate">
        <input type="submit" value="OK">
</pre></p></form></body></html>
```

view_details.html

```
<html>
<body bgcolor="#FFFFFF" text="#666666" link="#666699" vlink=
"#3399FF" alink="#996699">
<pre><br><br>
<p>
<font face="Monotype corsiva"><font color="#009999"><font
size="5">View Details By Date</font>
</p><p>
<font face="monotype corsiva"><font size="5">
<a href ="view_bydate_bycenter.html" target ="showframe">By Center
</a><br><br>
<a href ="view_bydate_byfarmer.html" target ="showframe">By Farmer
</a><br><br>
</pre></p></body></html>
```

Center_Add_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Center_Add_Servlet extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServletResponse
response)throws IOException, ServletException
        {
                ResultSet rset = null;
                Connection conn = null;
                Statement stmt = null;
                PreparedStatement pres = null;
                try
                {       Class.forName("oracle.jdbc.driver.OracleDriver");
                        String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
                        conn = DriverManager.get Connection (str,"system"
"tiger");
                        PrintWriter out = response.getWriter();
                        out.println("<html>");
                        int t_center_id=Integer.parseInt (request.getParameter
("center_id"));
                        String t_center_name=request.getParameter
("center_name");
```

```
                        String t_center_loc=request.getParameter ("center_loc");
                        String sql="insert into center values(?,?,?)";
                        pres=conn.prepareStatement(sql);
                        pres.setInt(1,t_center_id);
                        pres.setString(2,t_center_name);
                        pres.setString(3,t_center_loc);
                        pres.executeUpdate();
                        out.println("<body><font face='Monotype corsiva'><font
                color=#009999><font size=5><br><br><br><br><br> Center
                Id ~> "+t_center_id+"<br><br> Center Name ~>
                "+t_center_name+"<br><br> Center Location ~>
                "+t_center_loc);
                        conn.commit(); conn.close();
                        out.println("</font></body></html>");
                }
                catch(SQLException e)
                {
                        PrintWriter out = response.getWriter();
                        e.printStackTrace(); out.println("<html>");
                        out.println("<body><font face='Monotype corsiva'><font
        color=#009999><font size=5><br><br><br><br> Duplicate Center ID
        </font></body></html>");
                        System.out.println(e);
                } catch (Exception e){}
            }
        }
```

Center_Remove_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Center_Remove_Servlet extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServlet Response
response)throws IOException, ServletException
        {       Connection conn = null;
                Statement stmt = null;
                int r,r1,r2;
                try
                {       Class.forName("oracle.jdbc.driver.OracleDriver");
                        String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
                        conn = DriverManager.getConnection
(str,"system","tiger");
                        PrintWriter out = response.getWriter();
                        int t_center_id=Integer.parseInt (request.get
Parameter("center_id"));
                        out.println("<html>");
                        out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br>");                                String
sql="delete from center where center_id=?";
                        PreparedStatement pres=conn.prepareStatement(sql);
```

```
                    pres.setInt(1,t_center_id);
                    r = pres.executeUpdate();
                    String sql1="delete from farmer where center_id=?";
                    PreparedStatement pres1=conn.prepareStatement (sql1);
                    pres1.setInt(1,t_center_id);
                    r1 = pres1.executeUpdate();
                    String sql2="delete from procure where center_id=?";
                    PreparedStatement pres2=conn.prepareStatement (sql2);
                    pres2.setInt(1,t_center_id);
                    r2 = pres2.executeUpdate();
                    if(r==0&&r1==0&&r2==0)
                            out.println("<br><br>Invalid Center ID");
                    else
                            out.println("<br><br> Center
Removed</body></html>");
                    conn.commit(); conn.close();
                }
                catch(SQLException e)
                {       PrintWriter out = response.getWriter();
                        e.printStackTrace(); out.println("<html>");
                        out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br><br><br> Center ID not available
</font></body></html>");
                        System.out.println(e);
                }catch (Exception e){}
        }
}
```

Center_View_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Center_View_Servlet extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServletResponse
response)throws IOException, ServletException
        {
                ResultSet rset = null;
                Connection conn = null;
                Statement stmt = null;
                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                        String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
                        conn = DriverManager.getConnection
(str,"system","tiger");
                        PrintWriter out = response.getWriter();
                        int t_center_id=Integer.parseInt
(request.getParameter("center_id"));
                        out.println("<html>");
```

```
                    out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br>");                              String
sql="select * from center where center_id=?";
                    PreparedStatement pres=conn.prepareStatement(sql);
                    pres.setInt(1,t_center_id);
                    rset = pres.executeQuery();
                    int CENTRE_ID_COLUMN_INDEX = 1;
                    int CENTER_NAME_COLUMN_INDEX = 2;
                    int CENTER_LOC_COLUMN_INDEX = 3;
                    boolean flag=false;
                    while (rset.next())
                    {
                            flag=true;
                            int l_center_id = rset.getInt (
CENTRE_ID_COLUMN_INDEX );
                            String l_center_name = rset.getString (
CENTER_NAME_COLUMN_INDEX );
                            String l_center_loc = rset.getString (
CENTER_LOC_COLUMN_INDEX );
                            out.println("<br><br> Center Id ~> "
+l_center_id+"<br><br> Center Name ~> "+l_center_name+"<br><br>
Center Location ~> "+l_center_loc+"<br>");
                    }
                    if(!flag)
                    out.println("<br><br> Invalid Center ID");
                    out.println("</font></body></html>");
                }
```

```
		catch(SQLException e)
		{
			PrintWriter out = response.getWriter();
			e.printStackTrace();
			out.println("<html>");
			out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br><br><br> Center ID not available
</font></body></html>");
			System.out.println(e);
		}
		catch (Exception e)
		{
		}
	}
}
```

Compare_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Compare_Servlet extends HttpServlet
{
	public void doGet(HttpServletRequest request,HttpServlet Response
response)throws IOException, ServletException
```

```
    {
        ResultSet rset = null;
        Connection conn = null;
        Statement stmt = null;
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
            conn =
DriverManager.getConnection(str,"system","tiger");
            PrintWriter out = response.getWriter();
            int t_center_id=Integer.parseInt
(request.getParameter("center_id"));
            out.println("<html>");
            out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5>");
            String sql1="select c.center_id,c.fat,c.snf,c1.fat1,c1.snf1
from compare c, compare1  c1 where c.fat=c1.fat1 AND c.snf=c1.snf1 AND
c.center_id=?";
            PreparedStatement pres1=conn.prepareStatement(sql1);
            pres1.setInt(1,t_center_id);
            rset = pres1.executeQuery();
            int l_center_id; boolean flag=false;
            double l_fat,l_fat1; double l_snf,l_snf1;
            while (rset.next())
            {       flag=true;
                l_center_id = rset.getInt ( 1 );
```

```
                                        l_fat = rset.getDouble ( 2 );
                                        l_snf = rset.getDouble ( 3 );
                                        l_fat1 = rset.getDouble ( 4 );
                                        l_snfl = rset.getDouble ( 5 );
                                        out.println("<br><br>Values are equal<br><br>");
                                        out.println("Center Id ~> "
+l_center_id+"<br><br><br><br>Fat (Center) ~> "+l_fat+"<br><br>Snf
(Center) ~> "+l_snf+"<br><br><br><br>Fat (Dairy) ~>
"+l_fat1+"<br><br>Snf (Dairy) ~> "+l_snfl);
                              }
                              if(flag==false)
                              {
                                        out.println("<br><br>Values not equal");
                                        String sql2="insert into result values(?)";
                                        PreparedStatement
pres2=conn.prepareStatement(sql2);
                                        pres2.setInt(1,t_center_id);
                                        pres2.executeUpdate();
                              }
                              conn.commit(); conn.close();
                              out.println("</font></body></html>");
                    }
                    catch(SQLException e)
                    {         System.out.println("connection not established"); }
                    catch (Exception e){}
          }
}
```

Connect_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Connect_Servlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServlet Response
response)throws IOException, ServletException
    {
            if(request.getParameter("button").equals("button1"))
            {
            String source = "1";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
            f1.write(buf);
            f1.close();
            RequestDispatcher
rd1=getServletContext().getRequestDispatcher("/connectivity.html");
            rd1.forward(request,response);
        }
        else if(request.getParameter("button").equals("button2"))
        {
            String source = "2";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
            f1.write(buf);
            f1.close();
            RequestDispatcher
rd2=getServletContext().getRequestDispatcher("/connectivity.html");
            rd2.forward(request,response);
        }
        else if(request.getParameter("button").equals("button3"))
        {
            String source = "3";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
            f1.write(buf);
            f1.close();
            RequestDispatcher
rd3=getServletContext().getRequestDispatcher("/connectivity.html");
            rd3.forward(request,response);
        }
        else if(request.getParameter("button").equals("button4"))
        {
            String source = "4";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
```

```
            f1.write(buf);
            f1.close();
            RequestDispatcher
rd4=getServletContext().getRequestDispatcher("/connectivity.html");
            rd4.forward(request,response);
        }
        else if(request.getParameter("button").equals("button5"))
        {
            String source = "5";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
            f1.write(buf);
            f1.close();
            RequestDispatcher
rd5=getServletContext().getRequestDispatcher("/connectivity.html");
            rd5.forward(request,response);
        }
        else if(request.getParameter("button").equals("button6"))
        {
            String source = "6";
            byte buf[] = source.getBytes();
            OutputStream f1 = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
            f1.write(buf);f1.close();
            RequestDispatcher
rd6=getServletContext().getRequestDispatcher("/connectivity.html");
```

```
                        rd6.forward(request,response);
                }
                else if(request.getParameter("button").equals("button7"))
                {
                        String source = "7";
                        byte buf[] = source.getBytes();
                        OutputStream fl = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
                        fl.write(buf);fl.close();
                        RequestDispatcher
rd7=getServletContext().getRequestDispatcher("/connectivity.html");
                        rd7.forward(request,response);
                }
                else if(request.getParameter("button").equals("button8"))
                {
                        String source = "8";
                        byte buf[] = source.getBytes();
                        OutputStream fl = new FileOutputStream("C:/apache-
tomcat-5.5.26/webapps/Project/WEB-INF/classes/File1.txt");
                        fl.write(buf);fl.close();
                        RequestDispatcher
rd8=getServletContext().getRequestDispatcher("/connectivity.html");
                        rd8.forward(request,response);
                }
        }
}
```

Farmer_Add_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Farmer_Add_Servlet extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServlet Response
response)throws IOException, ServletException
        {
                ResultSet rset = null, rset1 = null;
                Connection conn = null;
                Statement stmt = null;
                PreparedStatement pres = null, pres1 = null;
                try
                {

                        Class.forName("oracle.jdbc.driver.OracleDriver");
                        String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
                        conn = DriverManager.getConnection
(str,"system","tiger");
                        PrintWriter out = response.getWriter();
                        out.println("<html><body><font face='Monotype
corsiva'><font color=#009999><font size=5><br><br>");
```

```
                    int
t_center_id=Integer.parseInt(request.getParameter("center_id"));
                    int
t_farmer_id=Integer.parseInt(request.getParameter("farmer_id"));
                    String
t_farmer_name=request.getParameter("farmer_name");
                    String
t_farmer_addr=request.getParameter("farmer_addr");
                    String sql="insert into farmer
(center_id,farmer_id,farmer_name,farmer_addr) values(?,?,?,?)";
                    pres=conn.prepareStatement(sql);
                    pres.setInt(1,t_center_id);
                    pres.setInt(2,t_farmer_id);
                    pres.setString(3,t_farmer_name);
                    pres.setString(4,t_farmer_addr);
                    pres.executeUpdate();
                    int i;
                    String resdt="";
                    int CENTER_ID_COLUMN_INDEX=1;
                    int FARMER_ID_COLUMN_INDEX=2;
                    int FARMER_NAME_COLUMN_INDEX=3;
                    int FARMER_ADDR_COLUMN_INDEX=4;
                    int DOJ_COLUMN_INDEX=5;
                    String sql1="select * from farmer where farmer_id=?";
                    pres1=conn.prepareStatement(sql1);
                    pres1.setInt(1,t_farmer_id);
                    rset1 = pres1.executeQuery();
```

```
            while (rset1.next())
            {
                int l_center_id = rset1.getInt (
CENTER_ID_COLUMN_INDEX );
                int l_farmer_id = rset1.getInt (
FARMER_ID_COLUMN_INDEX );
                String l_farmer_name = rset1.getString (
FARMER_NAME_COLUMN_INDEX );
                String l_farmer_addr = rset1.getString (
FARMER_ADDR_COLUMN_INDEX );
                Date l_doj = rset1.getDate (
DOJ_COLUMN_INDEX );
                String str1=l_doj.toString();
                String s[]= str1.split("-");
                for(i=0;i<s.length;i++)
                    System.out.println (s[i]);
                resdt=resdt.concat(s[2]).concat("-
").concat(s[1]).concat("-").concat(s[0]);
                out.println("<br> Centre Id ~>
"+l_center_id+"<br> Farmer Id ~> "+l_farmer_id+"<br> Farmer Name ~>
"+l_farmer_name+"<br> Farmer Address ~> "+l_farmer_addr+"<br> Date
of Joining ~> "+resdt);

            }
                out.println("<br><br></font></body></html>");
        conn.commit();
            conn.close();
```

```
                    out.println("</font></body></html>");
        }
                catch(SQLException e)
                {
                    PrintWriter out = response.getWriter();
                    e.printStackTrace();
                    out.println("<html>");
                    out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br><br><br> Duplicate Farmer ID
</font></body></html>");
                    System.out.println(e);
                }
                catch (Exception e)
                {
                }
        }
}
```

Farmer_Remove_Servlet.java

```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
```

```
public class Farmer_Remove_Servlet extends HttpServlet
{
    public void doGet(HttpServletRequest request,HttpServlet Response
response)throws IOException, ServletException
    {
        Connection conn = null;
        Statement stmt = null;
        int r,r1;
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
            conn = DriverManager.getConnection
(str,"system","tiger");
            PrintWriter out = response.getWriter();
            int
t_farmer_id=Integer.parseInt(request.getParameter("farmer_id"));
            out.println("<html>");
            out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br>");                    String
sql="delete from farmer where farmer_id=?";
            PreparedStatement pres=conn.prepareStatement(sql);
            pres.setInt(1,t_farmer_id);
            r = pres.executeUpdate();
            String sql1="delete from procure where farmer_id=?";
            PreparedStatement pres1=conn.prepareStatement(sql1);
            pres1.setInt(1,t_farmer_id);
            r1 = pres1.executeUpdate();

            if(r==0&&r1==0)
                out.println("<br><br> Invalid Farmer ID");
            else
                out.println("<br><br> Farmer Removed");

            out.println("</font></body></html>");
            conn.commit();
            conn.close();
        }
        catch(SQLException e)
        {
            PrintWriter out = response.getWriter();
            e.printStackTrace();
            out.println("<html>");
            out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br><br><br> Farmer ID not available
</font></body></html>");
            System.out.println(e);
        }
        catch (Exception e)
        {
        }
    }
}
```

Farmer_View_Servlet.java

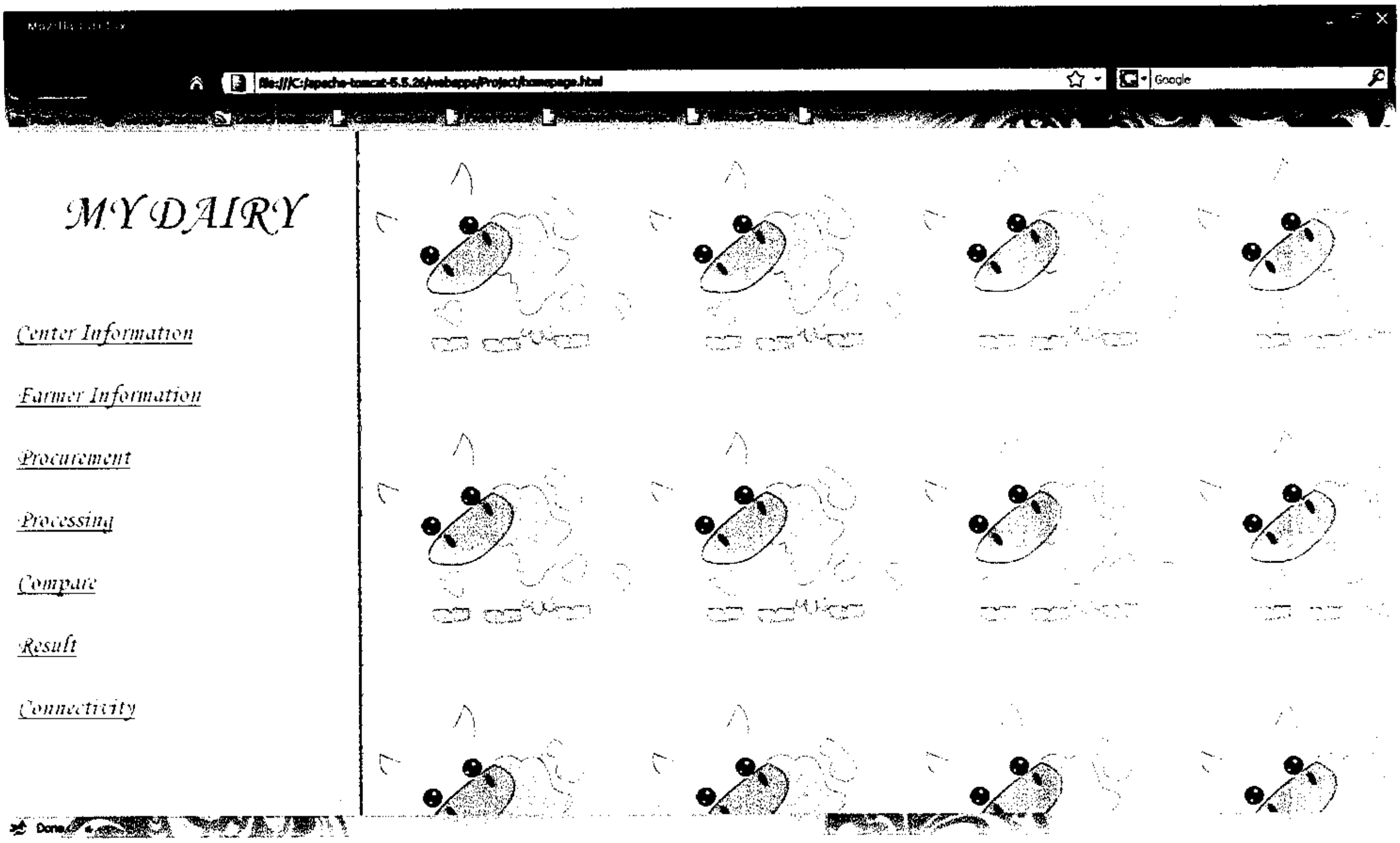
```
package s1;
import java.sql.*;
import java.lang.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Farmer_View_Servlet extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServletResponse
response)throws IOException, ServletException
        {

                ResultSet rset = null;
                Connection conn = null;
                Statement stmt = null;

                try
                {
                        Class.forName("oracle.jdbc.driver.OracleDriver");
                        String str = "jdbc:oracle:thin:@127.0.0.1:1521:orcl";
                        conn =
DriverManager.getConnection(str,"system","tiger");

                        PrintWriter out = response.getWriter();
                        int
t_farmer_id=Integer.parseInt(request.getParameter("farmer_id"));
```

```
            out.println("<html>");

            out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br>");

            String sql="select * from farmer where farmer_id=?";
            PreparedStatement pres=conn.prepareStatement(sql);
            pres.setInt(1,t_farmer_id);
            rset = pres.executeQuery();

            int i;
            String resdt="";
            int CENTER_ID_COLUMN_INDEX = 1;
            int FARMER_ID_COLUMN_INDEX = 2;
            int FARMER_NAME_COLUMN_INDEX = 3;
            int FARMER_ADDR_COLUMN_INDEX = 4;
            int DOJ_COLUMN_INDEX = 5;

            boolean flag=false;
            while (rset.next())
            {
                flag=true;
                int l_center_id = rset.getInt (
CENTER_ID_COLUMN_INDEX );
                int l_farmer_id = rset.getInt (
FARMER_ID_COLUMN_INDEX );
```

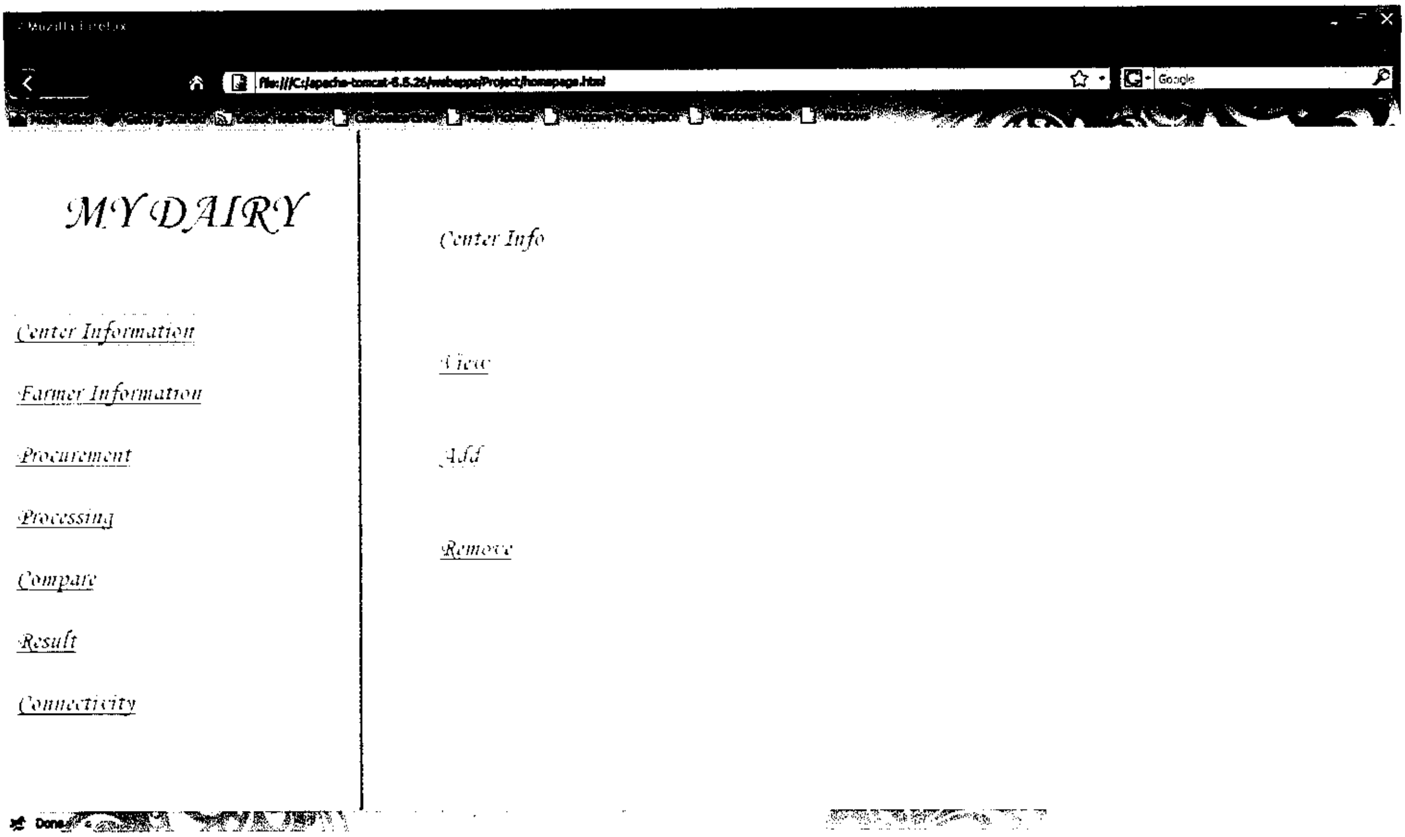
```
                    PrintWriter out = response.getWriter();
                    e.printStackTrace();
                    out.println("<html>");
                    out.println("<body><font face='Monotype corsiva'><font
color=#009999><font size=5><br><br><br><br> Farmer not available
</font></body></html>");
                    System.out.println(e);
            }
            catch (Exception e)
            {
            }
      }
}
```

SAMPLE OUTPUT

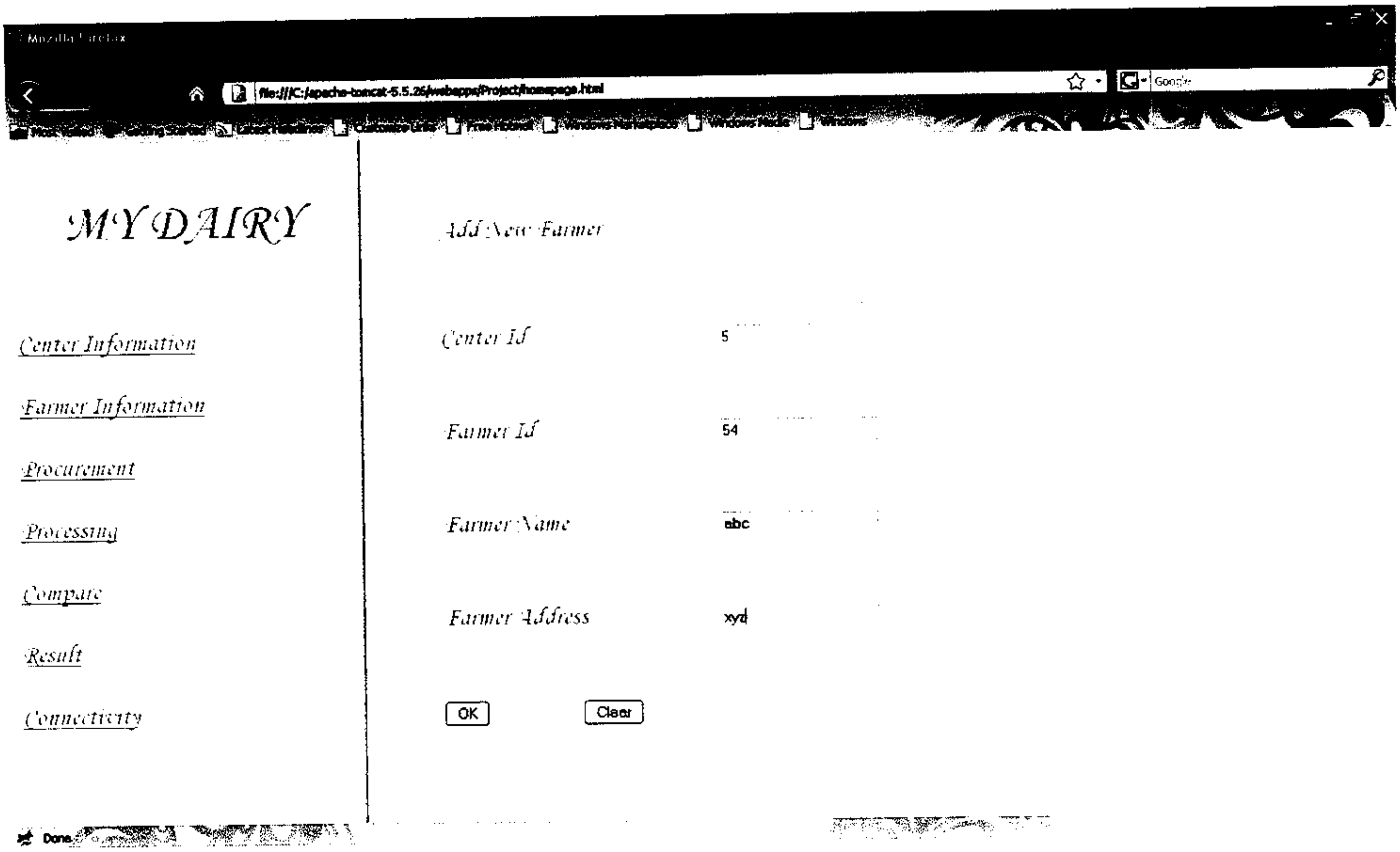
The homepage of the project provides information about the various links available in the website.



The information about the center is available in Center Information Link. The details namely the center id, name of the center and its location can be viewed. The system administrator can add new center, edit information of the existing center and can also remove a particular center.



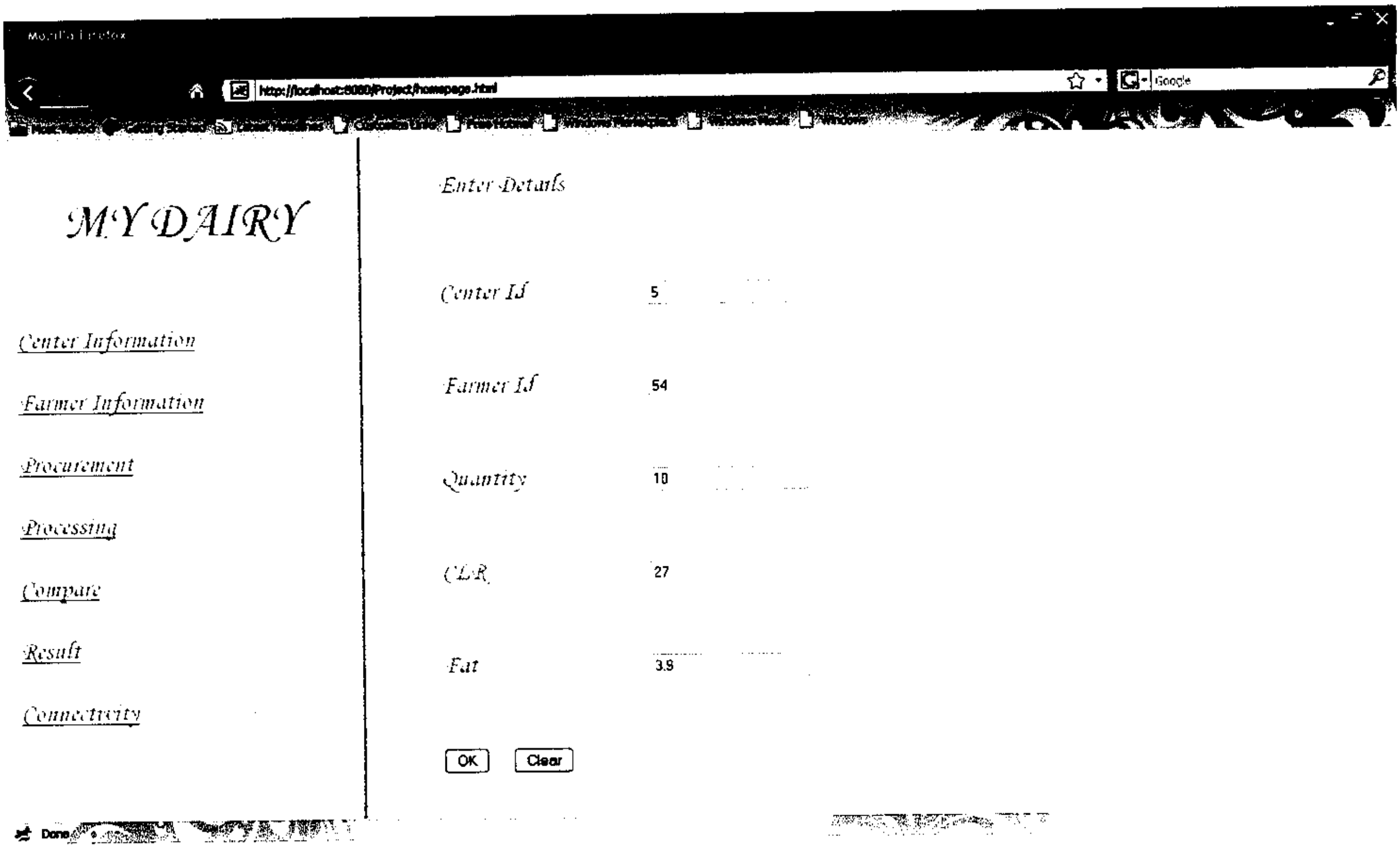
A new farmer is added to a center. A farmer can be added by giving a valid farmer id, the center's id to which he pours the milk farmer's name and address.



Add New Farmer

Center Id	5
Farmer Id	54
Farmer Name	abc
Farmer Address	xyd

The procurement details are entered at the collection center. The values of Fat and CLR of milk are entered. In addition to the quality parameters of milk, the quantity of milk is also entered. Rate per litre and the total rate for the quantity of milk poured by the farmer are displayed as the result.



Enter Details

Center Id 5

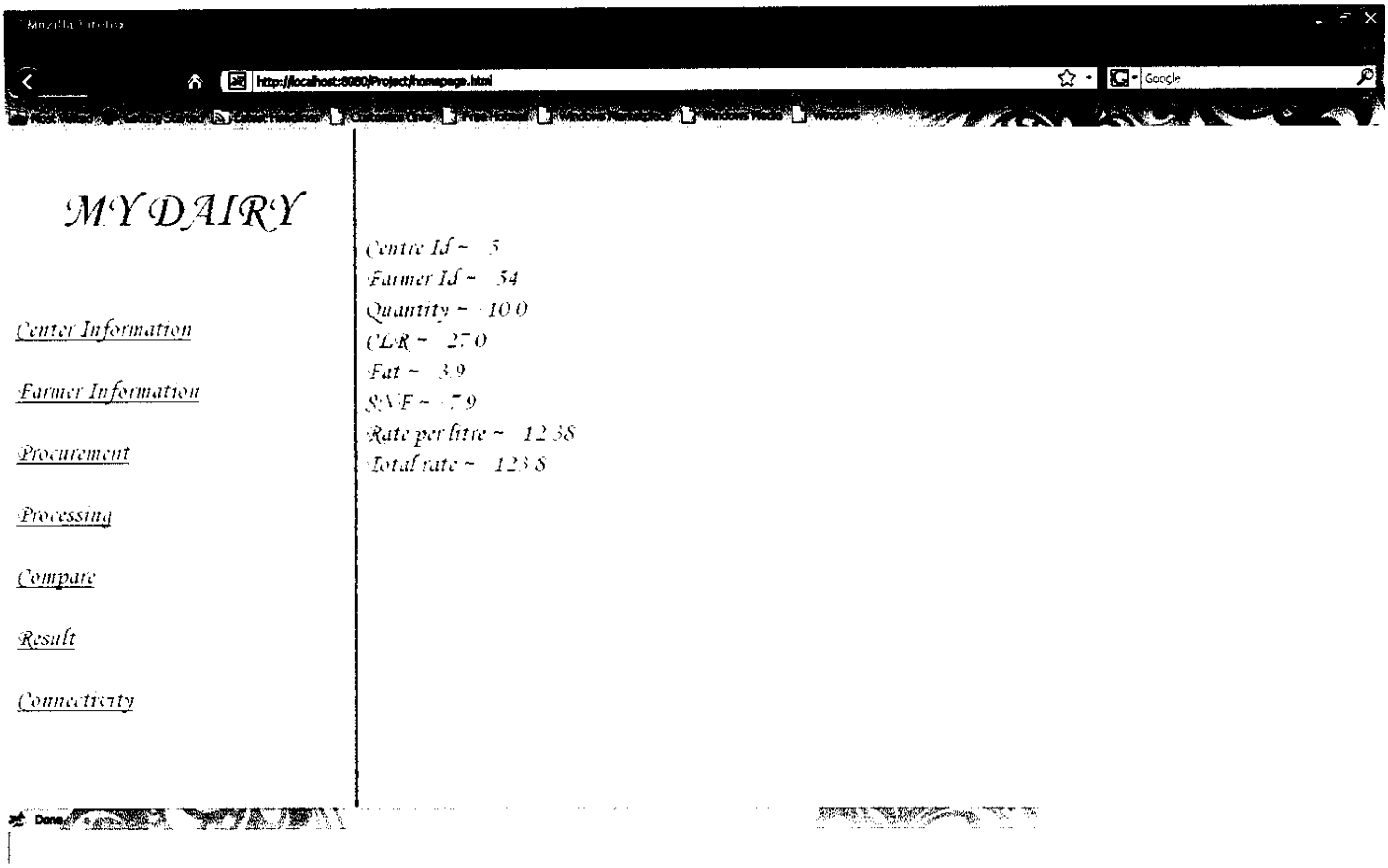
Farmer Id 54

Quantity 10

CLR 27

Fat 3.9

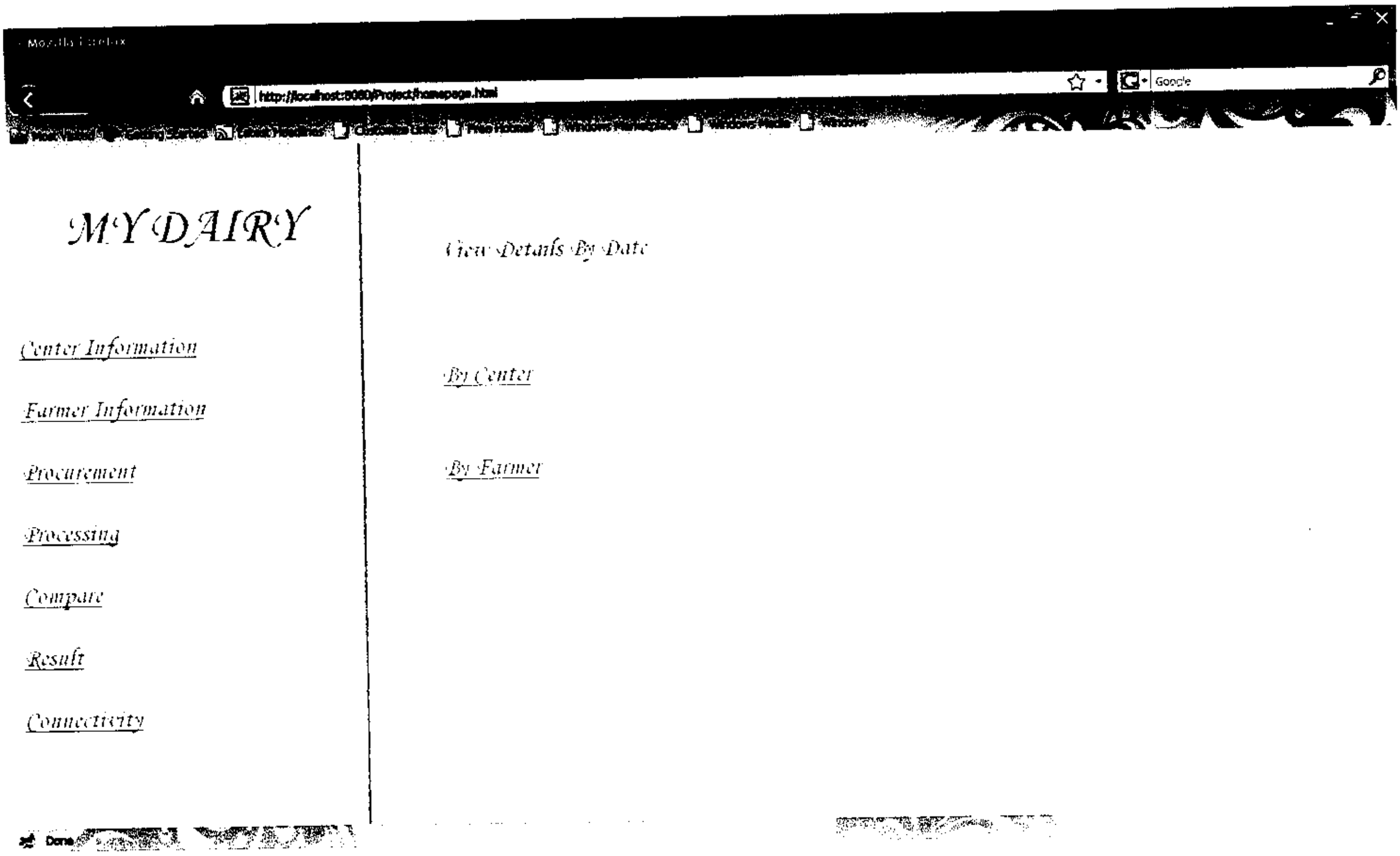
Procurement enter details resulting page.



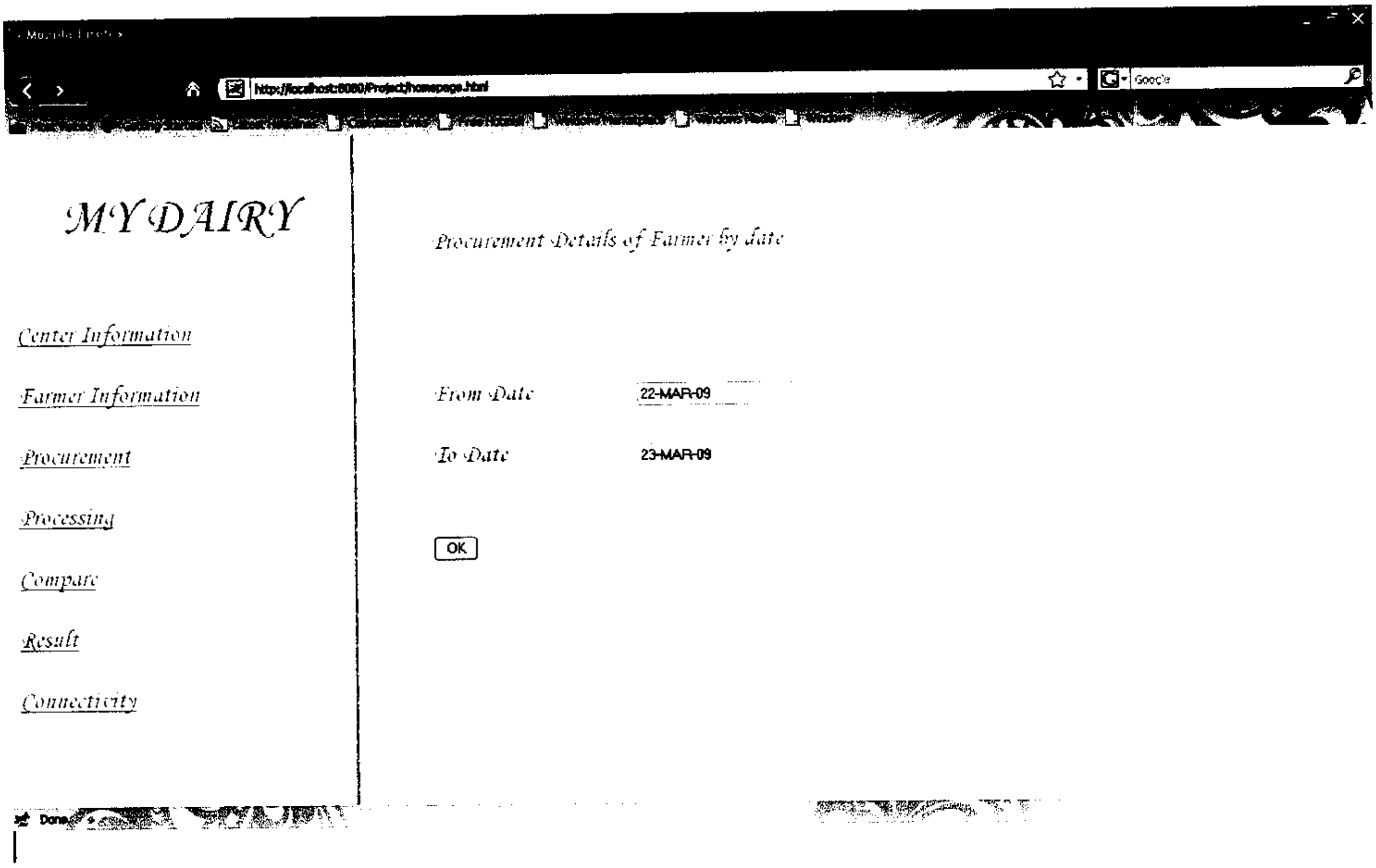
Centre Id ~ 5
Farmer Id ~ 54
Quantity ~ 10.0
CLR ~ 27.0
Fat ~ 3.9
SNF ~ 7.9
Rate per litre ~ 12.38
Total rate ~ 123.8

The Procurement Details can be viewed

- center wise
- farmer wise



To view the procurement details, ‘to’ and ‘from’ date is to be entered.

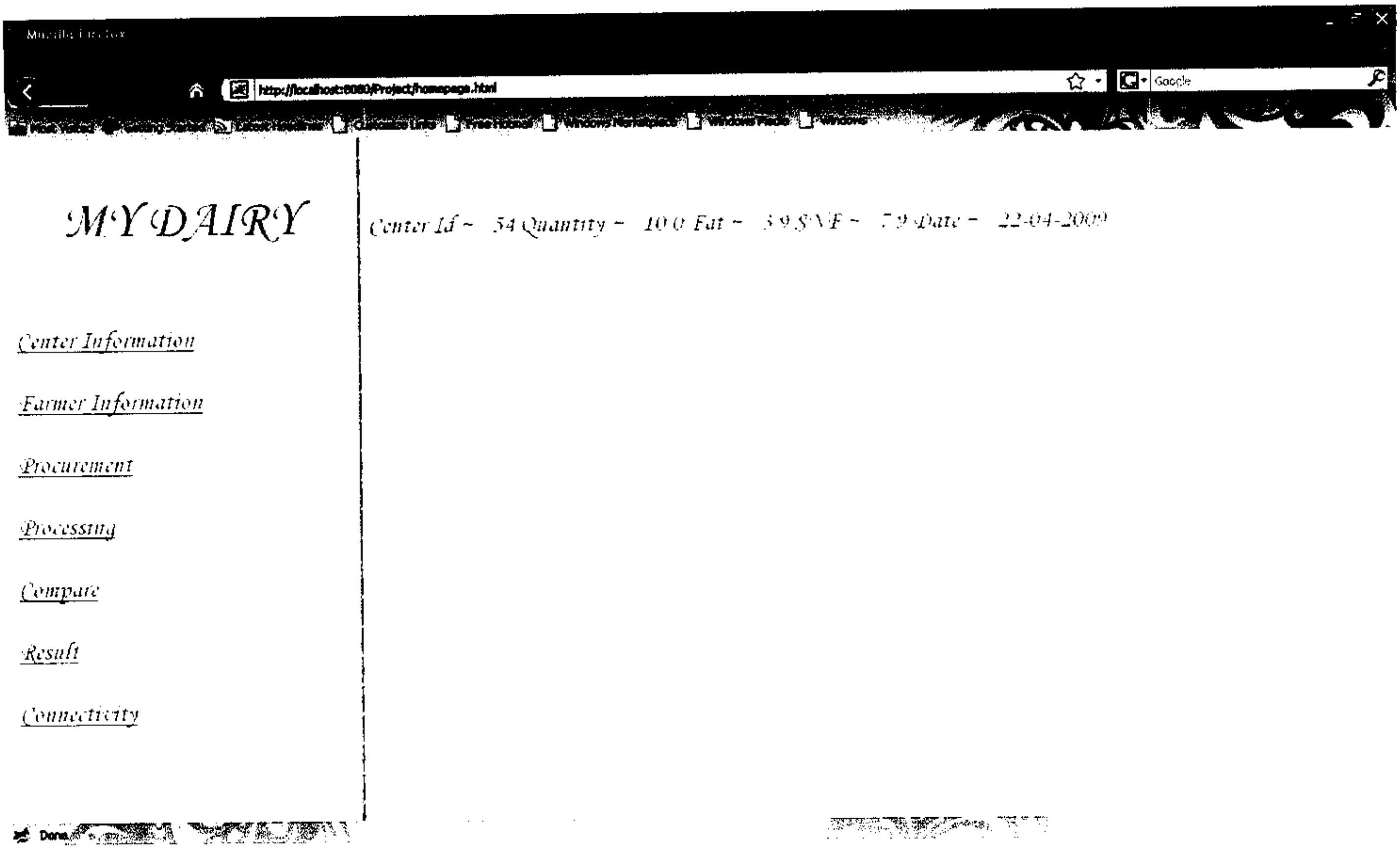


Procurement Details of Farmer by date

From Date 22-MAR-09

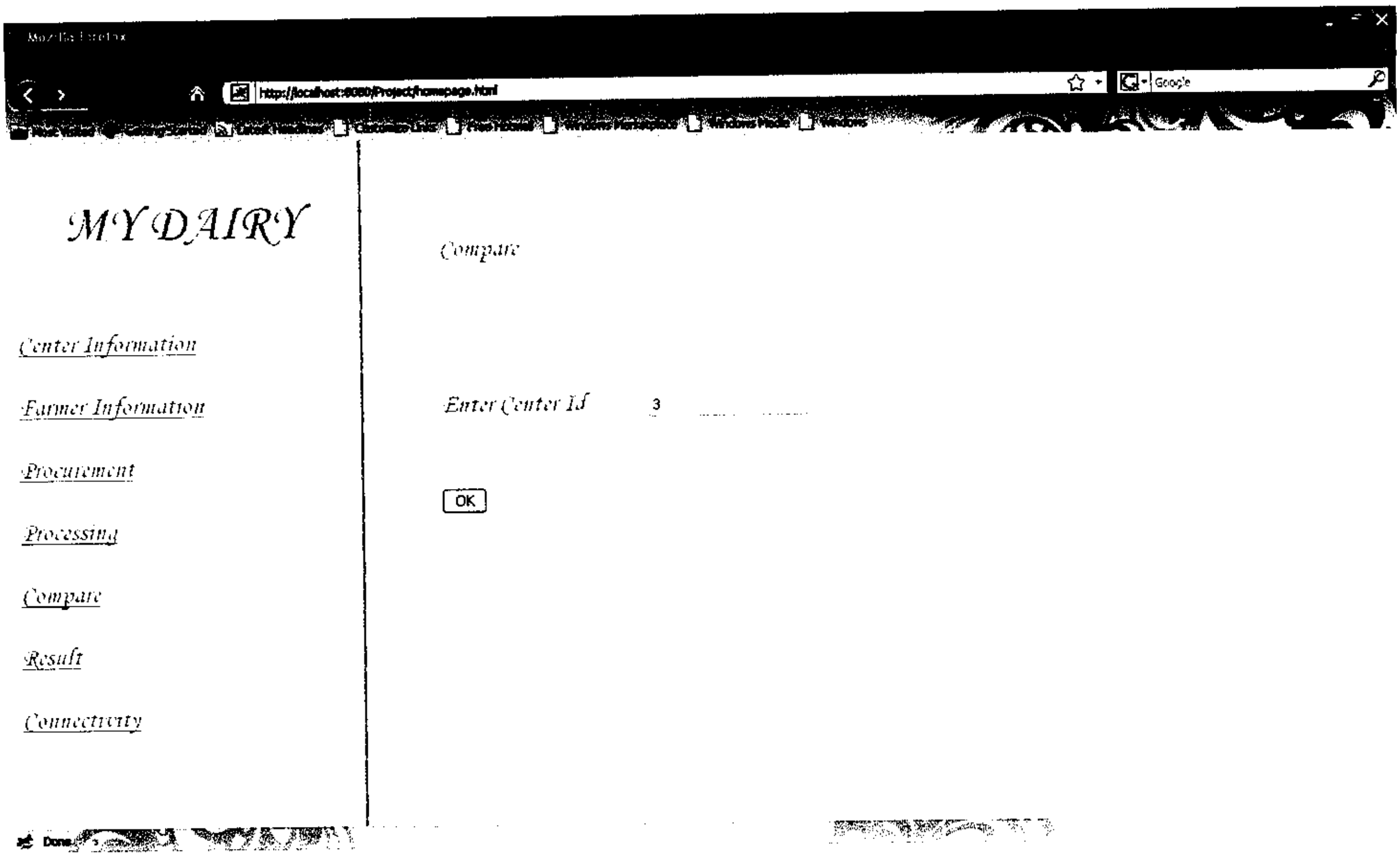
To Date 23-MAR-09

After entering the 'to' and 'from' date, the results are obtained from the database and displayed.



Center Id - 54 Quantity - 10.0 Fat - 3.9 SNF - 7.9 Date - 22-04-2009

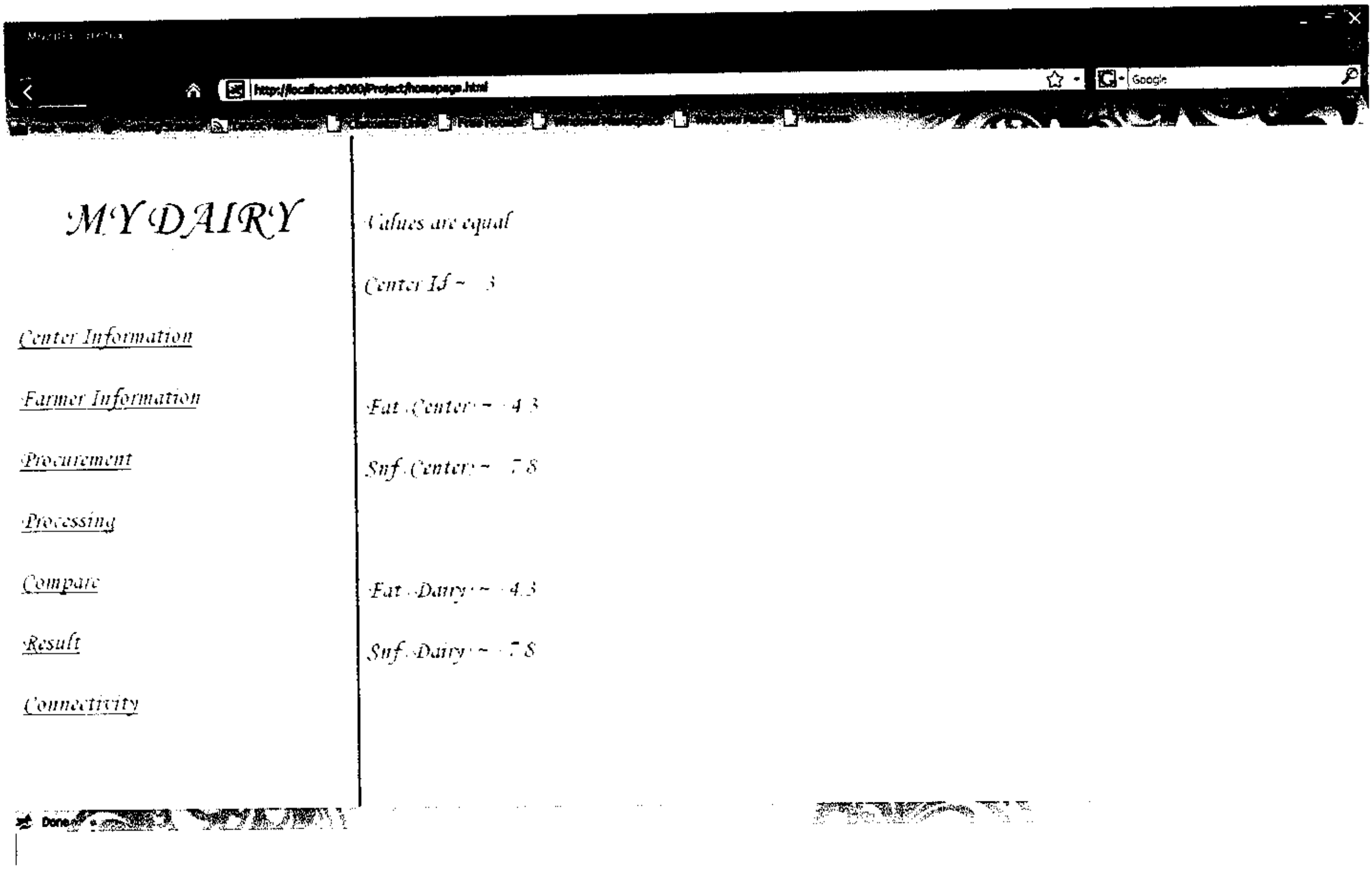
To compare the values obtained from the collection center with those from the dairy for a center, the center id should be specified.



Compare

Enter Center Id 3

The comparison result is displayed.



Values are equal

Center Id ~ 3

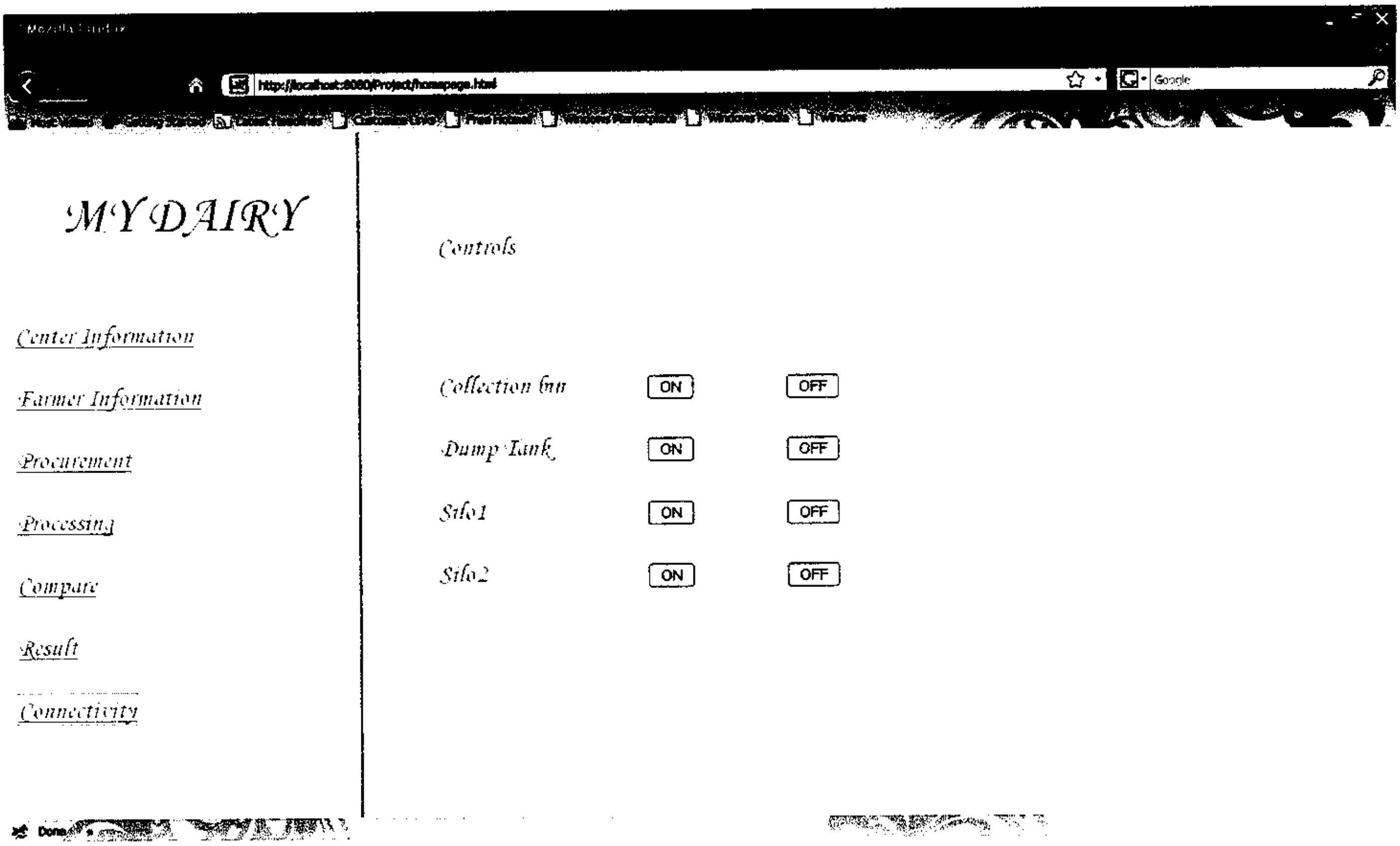
Fat Center ~ 4.3

Snf Center ~ 7.8

Fat Dairy ~ 4.3

Snf Dairy ~ 7.8

This page contains control buttons which help in switching on and switching off the pumps. Thus controlling of milk flow in dairy is made possible.



REFERENCES

REFERENCES

1. Aneja, R.P. and Banerjee, A.K. 'Technology of Indian Milk Products'.
2. National Dairy Development Board 'Milk Processing Manual'.
3. Schildt Herbert (2002) 'Java 2 : The Complete Reference, Fifth Edition'.
4. Tetra Pak Processing Systems 'Dairy Processing Handbook'.