# Introduction

Manufacturing environment and how they are applied to product design and manufacture. Emphasis is placed on the need for a total approach to product development, in particular the interface between the various functions of the design and make process and the use of simultaneous engineering.

Learners will start by investigating how CAM systems are used to increase the profitability of a business by reducing manufacturing costs, improving quality and being more responsive to customer needs. This is followed by a look at how simultaneous engineering is used to bring together the many functions of manufacturing business so that there is a team-based, multi-disciplinary approach to problem solving. Learners then investigate how simultaneous engineering can be used to meet the demands of a customer-driven economy where people expect an off-the shelf service for customised products.

The unit also covers how newly designed components are modelled using threedimensionalCAD software so that their functionality can be assessed and any errors corrected before the machining process is simulated using CAM software. Cutting metal is expensive and getting it right first time is a crucial aspect of economic manufacture.

Learners will investigate how manufacturing processes can be automated by using industrial robots to move materials and components between the machine tools and the workstations that make up a flexible manufacturing system (FMS). Finally, learners will be given the specification for a component, use CAD software to design it and use CAM software to produce a set of instructions for downloading to machine tool which could be used to make it.

**1.1 Problem Statement**

A **computer** is a general purpose device that can be programmed to carry out a finite set of arithmetic or logical operations. Since a sequence of operations can be readily changed, the computer can solve more than one kind of problem.

Conventionally, a computer consists of at least one processing element, typically a [central processing unit](http://en.wikipedia.org/wiki/Central_processing_unit) (CPU) and some form of [memory](http://en.wikipedia.org/wiki/Memory_%28computers%29). The processing element carries out arithmetic and logic operations, and a sequencing and control unit that can change the order of operations based on stored information. Peripheral devices allow information to be retrieved from an external source, and the result of operations saved and retrieved.

The first electron digital computers were developed between 1940 and 1945. Originally they were the size of a large room, consuming as much power as several hundred modern personal (PCs). In this era [mechanical](http://en.wikipedia.org/wiki/Mechanical_computer)  [analog computers](http://en.wikipedia.org/wiki/Analog_computers) were used for military applications.

Modern computers based on [integrated circuits](http://en.wikipedia.org/wiki/Integrated_circuit) are millions to billions of times more capable than the early machines, and occupy a fraction of the space. Simple computers are small enough to fit into [mobile devices](http://en.wikipedia.org/wiki/Mobile_device), and [mobile computers](http://en.wikipedia.org/wiki/Mobile_computing) can be powered by small [batteries](http://en.wikipedia.org/wiki/Battery_%28electricity%29). Personal computers in their various forms are [icons](http://en.wikipedia.org/wiki/Icon) of the [Information Age](http://en.wikipedia.org/wiki/Information_Age) and are what most people think of as “computers.” However, the [embedded computers](http://en.wikipedia.org/wiki/Embedded_system) found in many devices from [MP3 players](http://en.wikipedia.org/wiki/Digital_audio_player) to [fighter aircraft](http://en.wikipedia.org/wiki/Fighter_aircraft) and from toys to [industrial robots](http://en.wikipedia.org/wiki/Industrial_robot) are the most numerous.

The first use of the word “computer” was recorded in 1613 in a book called “The young mans gleanings” by English writer Richard Braithwait *I have read the truest computer of Times, and the best Arithmetician that euer breathed, and he reduceth thy dayes into a short number*. It referred to a person who carried out calculations, or computations, and the word continued with the same meaning until the middle of the 20th century. From the end of the 19th century the word began to take on its more familiar meaning, a machine that carries out computations.

**1.2 Existing System**

A personal computer (PC) is a general-purpose [computer](http://en.wikipedia.org/wiki/Computer), whose size, capabilities, and original sale price makes it useful for individuals, and which is intended to be operated directly by an [end-user](http://en.wikipedia.org/wiki/End-user) with no intervening computer operator. This contrasted with the batch processing or [time-sharing](http://en.wikipedia.org/wiki/Time-sharing_system) models which allowed larger, more expensive [minicomputer](http://en.wikipedia.org/wiki/Minicomputer) and [mainframe](http://en.wikipedia.org/wiki/Mainframe_computer) systems to be used by many people, usually at the same time. Large data processing systems require a full-time staff to operate efficiently.

Software applications for most personal computers include, but are not limited to, [word processing](http://en.wikipedia.org/wiki/Word_processing), [spreadsheets](http://en.wikipedia.org/wiki/Spreadsheets), [databases](http://en.wikipedia.org/wiki/Databases), [Web browsers](http://en.wikipedia.org/wiki/Web_browser) and [e-mail](http://en.wikipedia.org/wiki/E-mail) clients, [digital media](http://en.wikipedia.org/wiki/Digital_media) playback, [games](http://en.wikipedia.org/wiki/Personal_computer_game), and myriad personal productivity and special-purpose software applications. Modern personal computers often have connections to the [Internet](http://en.wikipedia.org/wiki/Internet), allowing access to the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web) and a wide range of other resources. Personal computers may be connected to a [local area network](http://en.wikipedia.org/wiki/Local_area_network) (LAN), either by a cable or a wireless connection. A personal computer may be a [desktop computer](http://en.wikipedia.org/wiki/Desktop_computer) or a [laptop](http://en.wikipedia.org/wiki/Laptop), [tablet](http://en.wikipedia.org/wiki/Tablet_computer), or a [handheld PC](http://en.wikipedia.org/wiki/Handheld_PC).

Early PC owners usually had to write their own programs to do anything useful with the machines, even lacking an [operating system](http://en.wikipedia.org/wiki/Operating_system). The very earliest microcomputers, equipped with a [front panel](http://en.wikipedia.org/wiki/Front_panel), required hand-loading of a [bootstrap](http://en.wikipedia.org/wiki/Booting) program to load programs from external storage ([paper tape](http://en.wikipedia.org/wiki/Paper_tape), cassettes, or eventually diskettes). Before very long, automatic booting from permanent read-only memory became universal.

Today's users have access to a wide range of [commercial software](http://en.wikipedia.org/wiki/Commercial_software) and [freeware](http://en.wikipedia.org/wiki/Freeware), which is provided in ready-to-run or ready-to-[compile](http://en.wikipedia.org/wiki/Compiler) form. Since the early 1990s, [Microsoft](http://en.wikipedia.org/wiki/Microsoft) operating systems and [Intel](http://en.wikipedia.org/wiki/Intel) hardware have dominated much of the personal computer market, first with [MS-DOS](http://en.wikipedia.org/wiki/MS-DOS) and then with the "[Wintel](http://en.wikipedia.org/wiki/Wintel)" (Windows + Intel) combination. Popular alternatives to Microsoft's [Windows](http://en.wikipedia.org/wiki/Windows) operating systems include [Apple's](http://en.wikipedia.org/wiki/Apple_Inc.) [OS X](http://en.wikipedia.org/wiki/OS_X) and the source Linux and [BSD](http://en.wikipedia.org/wiki/Berkeley_Software_Distribution) operating systems. [AMD](http://en.wikipedia.org/wiki/AMD) provides the major alternative to Intel's [central processing units](http://en.wikipedia.org/wiki/Central_processing_units). Applications and games for PCs are typically developed and distributed independently from the hardware or OS manufacturers, whereas software for many mobile phones and other portable systems is approved and distributed through a centralized online store.

DRAWBACK IN EXISTING SYSTEM

* Last 20 years ago the computer was very large in size
* Few years ago the monitor are used as desk tops and occupied space was large
* The process will also very slow

**1.3Proposed System**

Departments or individuals often bring applications into the company, or build their own using software like Access. Two types of problems result: the company is out of compliance because QA was not told and the system was not validated. In addition, the need to support a new system is sprung on IT. This procedure provides a computer system project proposal form and instructions. The form is used to assess if validation is required and the IT support needs before a new system is purchased or created. It also provides an early view into computer system implementations so that the organization can plan for the needed resources

**Advantages of Proposed System**

* Present generation computers was very fast in process
* Now a days monitor are using as LCD’s, LED’S.
* Computers occupied space is very less.

**2. REQUIREMENTS**

**2.1. Hardware requirements:**

**SERVER:**

Processor : Pentium IV

Speed : 1 GHz

Memory Capacity : 1 GB

Hard Disk Capacity : 40 GB

Monitor Make : HP

**CLIENT:**

Processor : Pentium IV

Speed : 1 GHz

Memory Capacity : 256 MB

Hard Disk Capacity : 20 GB

Monitor Make : HP

**2.2. Software Requirements:**

Operating System : Windows XP Professional

Web Server : Apache Tomcat Web Server

Database : Ms Access

Enterprise Application : J2EE, JBOSS

Implementation Architecture : MVE,

3 Tier using Servlets, JSP

Scripting Languages : Java Script

Programming Language : Java

**HTML:**

HTML is a computer language devised to allow website creation. These websites can then be viewed by anyone else connected to the Internet. It is relatively easy to learn, with the basics being accessible to most people in one sitting; and quite powerful in what it allows you to create. It is constantly undergoing revision and evolution to meet the demands and requirements of the growing Internet audience under the direction of the>>w3c, the organization charged with designing and maintaining the language.

The definition of HTML is Hyper Text Markup Language. Hyper Text is the method by which you move around on the web — by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not linear — i.e. you can go to any place on the Internet whenever you want by clicking on links — there is no set order to do things in.

HTML is a Language, as it has code-words and syntax like any other language.

HTML consists of a series of short codes typed into a text-file by the site author — these are the tags. The text is then saved as an Html file, and viewed through a [browser](http://www.yourhtmlsource.com/starthere/glossary.html#browser), like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form, hopefully rendering the page as the author had intended.

**HTML Tags:**

The tags are what separate normal text from HTML code. You might know them as the words between the <angle-brackets>. They allow all the code stuff like images and tables and stuff, just by telling your browser what to render on the page. Different tags will perform different functions. The tags themselves don’t appear when you view your page through a browser, but their effects do. The simplest tags do nothing more than apply formatting to some text, like this:

**Java Server Pages:**

Java Server Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.A Java Server Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.

**Use JSP:**

Java Server Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI.

Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.

JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.

**Advantages of JSP vs. Active Server Pages (ASP):**

The advantages of JSP are twofold. First, the dynamic part is written in Java, not Visual Basic or other MS specific language, so it is more powerful and easier to use. Second, it is portable to other operating systems and non-Microsoft Web servers.

**vs. Pure Servlets:**

It is more convenient to write (and to modify!) regular HTML than to have plenty of println statements that generate the HTML.

**vs. Server-Side Includes (SSI):**

SSI is really only intended for simple inclusions, not for "real" programs that use form data, make database connections, and the like.

**vs. JavaScript:**

JavaScript can generate HTML dynamically on the client but can hardly interact with the web server to perform complex tasks like database access and image processing etc.

**vs. Static HTML:**

Regular HTML, of course, cannot contain dynamic information.I would take you step by step to set up your environment to start with JSP. I'm assuming you have good hands on with Java Programming to proceed with learning JSP.If you are not aware of Java Programming Language then I would recommend to go through Java Tutorial to understand Java Programming.

**Setting up Java Development Kit:**

If you are running Windows and installed the SDK in C:\jdk1.5.0\_20, you would put the following line in your C:\autoexec.bat file.

set PATH=C:\jdk1.5.0\_20\bin;%PATH%

set JAVA\_HOME=C:\jdk1.5.0\_20

Alternatively, on Windows NT/2000/XP, you could also right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the PATH value and press the OK button.On Unix (Solaris, Linux, etc.), if the SDK is installed in /usr/local/jdk1.5.0\_20 and you use the C shell, you would put the following into your .cshrc file.

setenv PATH /usr/local/jdk1.5.0\_20/bin:$PATH

setenv JAVA\_HOME /usr/local/jdk1.5.0\_20

Alternatively, if you use an Integrated Development Environment (IDE) like Borland JBuilder, Eclipse, IntelliJ IDEA, or Sun ONE Studio, compile and run a simple program to confirm that the IDE knows where you installed Java.

**Setting up Web Server Tomcat:**

A number of Web Servers that support JavaServer Pages and Servlets development are available in the market. Some web servers are freely downloadable and Tomcat is one of them.

Apache Tomcat is an open source software implementation of the JavaServer Pages and Servlet technologies and can act as a standalone server for testing JSP and Servlets and can be integrated with the Apache Web Server. Here are the steps to setup Tomcat on your machine:

Download latest version of Tomcat from http://tomcat.apache.org/.

Once you downloaded the installation, unpack the binary distribution into a convenient location. For example in C:\apache-tomcat-5.5.29 on windows, or /usr/local/apache-tomcat-5.5.29 on Linux/Unix and create CATALINA\_HOME environment variable pointing to these locations.

Tomcat can be started by executing the following commands on windows machine:

%CATALINA\_HOME%\bin\startup.bat

or

C:\apache-tomcat-5.5.29\bin\startup.bat

Tomcat can be started by executing the following commands on Unix (Solaris, Linux, etc.) machine:

$CATALINA\_HOME/bin/startup.sh

Or

/usr/local/apache-tomcat-5.5.29/bin/startup.sh

After a successful startup, the default web applications included with Tomcat will be available by visiting http://localhost:8080/. If everything is fine then it should display following result:

**Tomcat Home page:**

Further information about configuring and running Tomcat can be found in the documentation included here, as well as on the Tomcat web site: http://tomcat.apache.org

Tomcat can be stopped by executing the following commands on windows machine:

The web server needs a JSP engine ie. container to process JSP pages.

**JSP Architecture:**

**JSP Processing:**

As with a normal page, your browser sends an HTTP request to the web server.The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with. jsp instead of .html.

The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code that implements the corresponding dynamic behavior of the page.

**JSP Life Cycle:**

A JSP life cycle can be defined as the entire process from its creation till the destruction which is similar to a servlet life cycle with an additional step which is required to compile a JSP into servlet.

**JSP Compilation:**

When a browser asks for a JSP, the JSP engine first checks to see whether it needs to compile the page. If the page has never been compiled, or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

Parsing the JSP.

Turning the JSP into a servlet.

Compiling the servlet.

**JSP Initialization:**

When a container loads a JSP it invokes the jspInit() method before servicing any requests. If you need to perform JSP-specific initialization, override the jspInit() method:

public void jspInit(){

// Initialization code...

}

Typically initialization is performed only once and as with the servlet init method, you generally initialize database connections, open files, and create lookup tables in the jspInit method.

**JSP Execution:**

This phase of the JSP life cycle represents all interactions with requests until the JSP is destroyed. Whenever a browser requests a JSP and the page has been loaded and initialized, the JSP engine invokes the \_jspService() method in the JSP.

The \_jspService () method takes an HttpServletRequest and an HttpServletResponse as its parameters as follows:

Void \_jspService (HttpServletRequest request,

HttpServletResponse response)

{

// Service handling code...

}

The \_jspService() method of a JSP is invoked once per a request and is responsible for generating the response for that request and this method is also responsible for generating responses to all seven of the HTTP methods i.e. GET, POST, DELETE etc.

**JSP Cleanup:**

The destruction phase of the JSP life cycle represents when a JSP is being removed from use by a container.

The jspDestroy () method is the JSP equivalent of the destroy method for servlets. Override jspDestroy when you need to perform any cleanup, such as releasing database connections or closing open files.

**Microsoft access:**

Microsoft Access is a database management program that is generally bundled with the Microsoft Office package. Although it is among the least used of the software bundle, Access provides very intuitive solutions for small business and corporate users who need a simple data management tool with deep compatibility and professional options.

Access is a database tool for gathering and understanding all your information—your phone numbers, inventory, guest lists, whatever you’re tracking—and providing a convenient way to enter, navigate, and report out your data.

**Overall Description:**

Overall description is used in logistical systems such as warehousing, transportation, distributorships and supply management chains, an inventory management system is an automated system that keeps track of inventory on a regular basis.

The system ensures that items marked for inventory and distribution are accounted for and that stock levels, quantities and costs are reasonable when requisitions for inventory are placed through the system by automatic or manual re-ordering. An inventory management system also keeps an audit trail of all transactions.

**3. MODULES**

The computerization of manufacturing unit have five module they are

* Administrator
* User
* Raw materials
* Products
* Manager

Example: we are manufacturing the computer

**Administrator module:**

Administrator will allow disallow of user, manager login products He will give the details of components What are the raw material want to use for the manufacture of the computer He will feed the list of raw materials & the list of component product range, cost & memory etc…He will concentrate on the product, manage group, manage user, user sessions, manager logs..

**Manager module:**

Develop a plan for lean **manufacturing** based on lean principles-create more value through the elimination of waste. If necessary, hire a lean expert to form and train an internal team. This team will study all of the production areas and make recom.The manager will check the day-to-day product work He will give the updates of raw material& product to userHe want to concentrate on the quality, cost of price, monitors delivery date .

**Client/user module:**

User will give the product requirements to the manager He will give deadline for product and

Select the product

**Raw material module:**

* Cpu
* Hard disk
* Ram
* Processer
* Key board
* Mouse
* Monitor
* Details entry of the raw materials

**Product module:**

* Product id
* Product name
* Product price
* Product dimension
* Product details entry
* Product delivery entry

**4. UML DIAGRAMS**

**UNIFIED MODELING LANGUAGE:**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + User Model View
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + Structural model view
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* Implementation Model View

In this the structural and behavioral as parts of the system are represented as they are to be built.

* Environmental Model View

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are:

* UML Analysis modeling, this focuses on the user model and structural model views of the system.
* UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

# 4.1 Use-case Diagram:

In software engineering, a use-case diagram in the Unified Modeling Language (UML) is a type of Behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use-cases.

****

# 4.2 Class Diagram:

Use cases and use case diagrams must at some point be translated into classes for eventual implementation. In UML, a class is represented by a rectangular box which in its most detailed representation is divided into three parts vertically. The name of the class is written in the uppermost partition of the box, followed by the class data members (called attributes in the parlance of UML) in the middle partition, followed by its member functions or methods (called operations in UML) in the lowest partition. The name of the class is shown in bold for a concrete class, and in italics for an abstract class.



**4.3 Sequence Diagram:**

A sequence diagram is a kind of in interaction diagram that shows how processes operate with one another and in what order. It is a construct of message sequence chart. A sequence diagram shows, as parallel vertical lines, different processes or objects that live simultaneously and, as horizontal arrows, the messages exchange between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



**4.4 Collaboration Diagram:**

A collaboration diagram, also called a communication diagram or interaction diagram. A collaboration diagram is easily represented by modeling objects in a system and representing the associations between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of these arrows. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. Hence, the elements of a collaboration diagram are essentially the same as that of a sequence diagram.

****

# 4.5 Activity Diagram:

Activity diagrams are diagram techniques showing workflows of stepwise activities and actions,with support for choice,iteration and concurrency.In the Unified Modeling Language,activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.An activity diagram shows the overall flow of control.

Administrator

Client

Manager

Enter raw material &product details

Login

Login

Login

Search product

Enter product delivery status

Re-Order raw materials

Enter delivery product details

Ordered product

Check delivery status

Update product details

Nenterenter raw materials and product details

Yes

start

Enter raw `materials receipt status

Product available

No

end

**4.6. Database Tables:**

**Registration:**

|  |  |
| --- | --- |
| **Username** | **Text** |
| **Password** | **Text** |
| **Firstname** | **Text** |
| **Lastname** | **Text** |
| **Gender** | **Text** |
| **Age** | **Number** |
| **Profession** | **Text** |
| **Phno** | **Number** |
| **Email** | **Text** |
| **Address** | **Text** |

**Login**

|  |  |
| --- | --- |
| **Username** | **Text** |
| **Password** | **Text** |

**5. CODE**

**HTML CODE for Main Page:**

<html>

<title>computer manufacturing </title>

</head>

<body bgcolor="sky blue"" text="#7FFF00" link="#FF0000" vlink="#000000" alink="#0000FF">

<hr color="#800000" size="8">

<p align="center"><font size="8">COMPUTERIZATION OF MANUFACTURING UNIT</font></p>

<hr color="#800000" size="8">

<div id="header" style="background-color:#00FFFF;"><h1><a href=registration.html style="text-decoration: none; font-weight:2000">

<center>Registration</center></a><br>

<a href=user.html style="text-decoration: none; font-weight: 800">

<center>user</center></a><br>

<a href=manager.html style="text-decoration: none; font-weight: 800">

<center>manager</center></a><br>

<a href=rawmaterials.html style="text-decoration: none; font-weight: 800">

<center>raw material</center></a><br>

<a href=product.html style="text-decoration: none; font-weight: 800">

<center>product</center></h1></a>

</font>

</body>

</html>

**Registration page:**

<html>

<head>

<title>Visa Processing System</title>

</head>

<body text="330000" bgcolor=#FFC2E0>

<center><form name="visa1" method="post" action="jvisa2.jsp">

<pre>

<h2>Computer Manfacturing Unit</h2>

Username :<input type="text" name="n1"><br>

Password :<input type="password" name="n2"><br>

firstname :<input type="text" name="n3"><br>

lastname :<input type="text" name="n4"><br>

gender :<input type="text" name="n5"><br>

Age :<input type="text" name="n6"><br>

profession :<input type="text" name="n7"><br>

telephoneno :<input type="text" name="n8"><br>

email :<input type="text" name="n9"><br>

address :<input type="text" name="n10"><br>

<input type="submit" value="Submit">

</pre>

</form></center>

</body>

</html>

**Login Page:**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1" page Encoding="ISO-8859-1"%>

<html>

<head>

<title>cmp</title>

</head>

<body text="330000" bgcolor=#FFC2E0>

<center>

<pre>

<font size=3 color="330000" face=dutch801 rmbt>

<form action=next.jsp>

<h2>User Login</h2>

username :<input type="text" name="n1"><br>

password :<input type="password" name="n2"><br>

<input type="submit" value="Submit">

</pre>

</form></center>

</body>

</html>

**Jsp Code for Registration**

%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@page import="java.sql.\*"%>

<html>

<head>

</head>

<body>

<%

String s1=request.getParameter("n1");

String s2=request.getParameter("n2");

String s3=request.getParameter("n3");

String s4=request.getParameter("n4");

String s5=request.getParameter("n5");

String s6=request.getParameter("n6");

int ph1 = Integer.parseInt(s6);

String s7=request.getParameter("n7");

String s8=request.getParameter("n8");

int d2 = Integer.parseInt(s8);

String s9=request.getParameter("n9");

String s10=request.getParameter("n10");

PreparedStatement ps=null;

Connection conn=null;

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

conn = DriverManager.getConnection("jdbc:odbc:anu");

ps=conn.prepareStatement("insert into reg values(?,?,?, ?, ?, ?, ?, ?, ?, ?)");

ps.setString(1, s1);

ps.setString(2, s2);

ps.setString(3, s3);

ps.setString(4, s4);

ps.setString(5, s5);

ps.setInt(6, ph1);

ps.setString(7, s7);

ps.setInt(8, d2);

ps.setString(9, s9);

ps.setString(10, s10);

ps.executeUpdate();

out.println("insertion done");

%>

</body>

</html>

**Jsp code for Login:**

%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<%@page import="java.sql.\*"%>

<html>

<head>

</head>

<body>

<%

String s1=request.getParameter("n1");

String s2=request.getParameter("n2");

PreparedStatement ps=null;

Connection conn=null;

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

conn = DriverManager.getConnection("jdbc:odbc:anu");

ps=conn.prepareStatement("insert into user1 values(?, ?)");

ps.setString(1, s1);

ps.setString(2, s2);

ps.executeUpdate();

out.println("insertion done");

%>

</body>

</html>

**6. TESTING**

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also Psychology of Testing.

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing with the intent of showing that a program works, but the intent should be to show that a program doesn’t work.

Testing is the process of executing a program with the intent of finding errors.

**Testing Objectives**:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we can say,

* Testing is a process of executing a program with the intent of finding an error.
* A successful test is one that uncovers an as yet undiscovered error.
* A good test case is one that has a high probability of finding error, if it exists.
* The tests are inadequate to detect possibly present errors.
* The software more or less confirms to the quality and reliable standards.

**LEVELS OF TESTING:**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are

Client Needs Acceptance Testing

Requirements System Testing

Design Integration Testing

Code Unit Testing

A series of testing is done for the proposed system before the system is ready for the user acceptance testing.

The steps involved in Testing are:

**Unit Testing:**

Unit testing focuses verification efforts on the smallest unit of the software design**,** the module**.** This is also known as “Module Testing”**.** The modules are tested separately**.** This testing carried out during programming stage itself**.** In this testing each module is found to be working satisfactorily as regards to the expected output from the module**.**

**Integration Testing:**

Data can be grossed across an interface**;** one module can have adverse efforts on another**.** Integration testing is systematic testing for construction the program structure while at the same time conducting tests to uncover errors associated with in the interface. The objective is to take unit tested modules and build a program structure**.** All the modules are combined and tested as a whole**.** Here correction is difficult because the isolation of cause is complicate by the vast expense of the entire program. Thus in the integration testing stop**,** all the errors uncovered are corrected for the text testing steps**.**

**System testing:**

System testing is the stage of implementation that is aimed at ensuring that the system works accurately and efficiently for live operation commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, then goal will be successfully achieved.

**Validation Testing:**

At the conclusion of integration testing software is completely assembled as a package, interfacing errors have been uncovered and corrected and a final series of software tests begins**,** validation test begins**.** Validation test can be defined in many ways**.** But the simple definition is that validation succeeds when the software function in a manner that can reasonably expected by the customer. After validation test has been conducted one of two possible conditions exists.

One is the function or performance characteristics confirm to specifications and are accepted and the other is deviation from specification is uncovered and a deficiency list is created. Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

**Output Testing:**

After performing validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated by the system under consideration. Here the output format is considered in two ways, one is on the screen and other is the printed format. The output format on the screen is found to be correct as the format was designed in the system designed phase according to the user needs. For the hard copy also the output comes as the specified requirements by the users. Hence output testing does not result any corrections in the system.

**User Acceptance Testing:**

User acceptance of a system is the key factor of the success of any system. The system under study is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required.

**Test Data:**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing after preparing the test data the system under study is tested using the test data. While testing the system by using the test data errors are again uncovered and corrected by using above testing steps and corrections are also noted from the future use.

**Testing:**

The testing done here was System Testing–checking whether the user requirements were satisfied. The code for the new system has been written completely using JSP as the coding language, HTML as the interface for front-end designing and Java Script for validating the client-side applications. The new system has been tested well with the help of the users and all the applications have been verified from every nook and corner of the user.

Although some applications were found to be erroneous these applications have been corrected before being implemented. The flow of the forms has been found to be very much in accordance with the actual flow of data.

**7. SCREEN SHOTS**

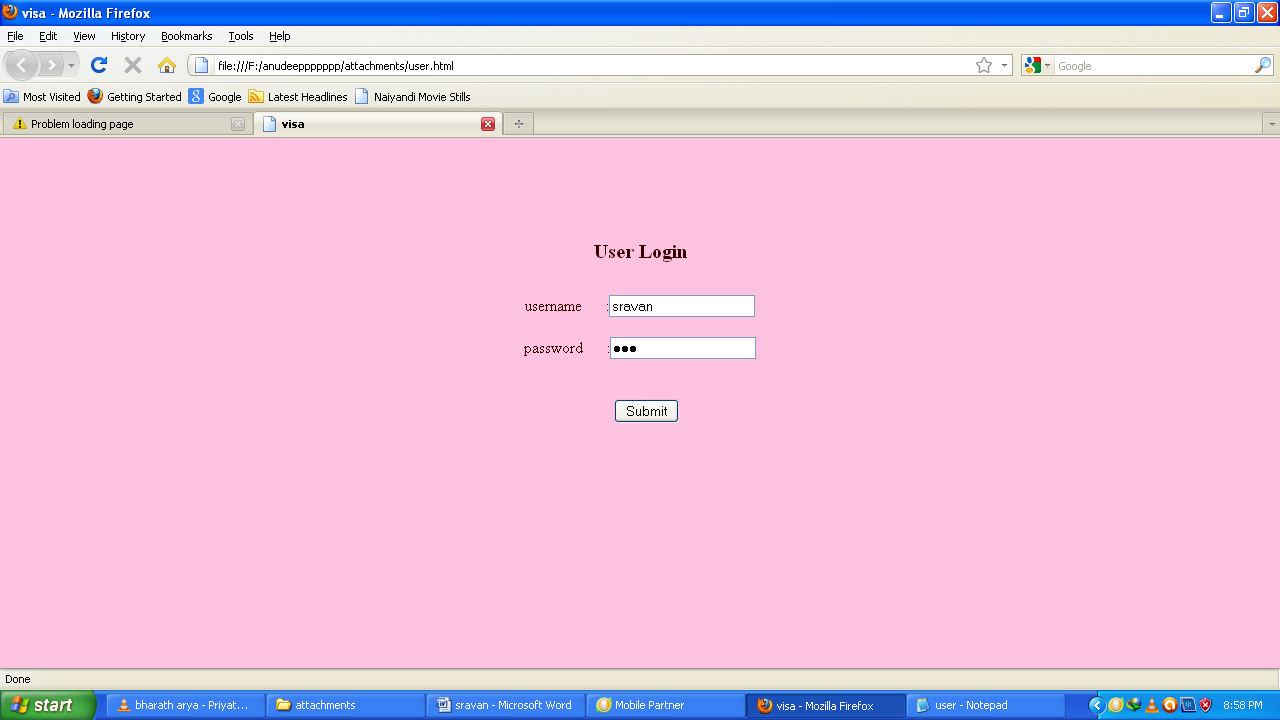
**Home page:**



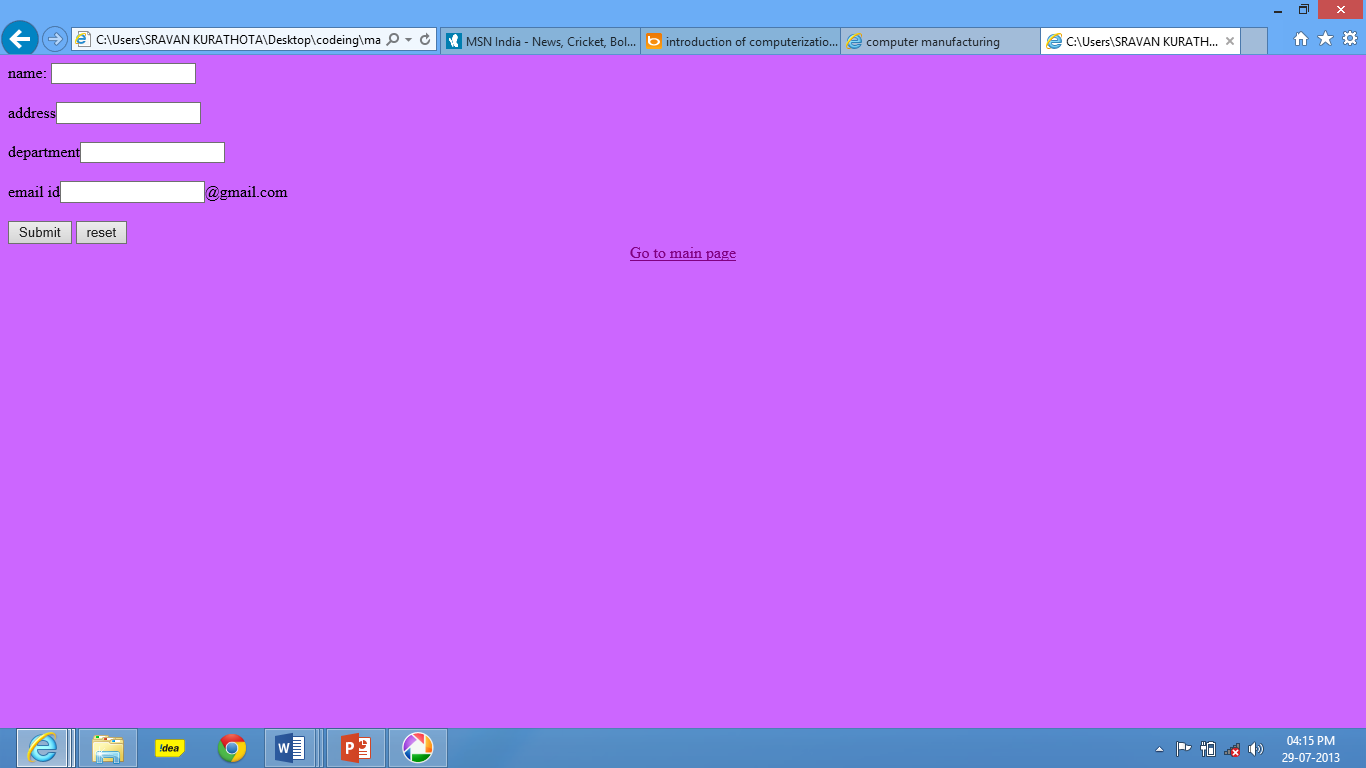
**Registration:**



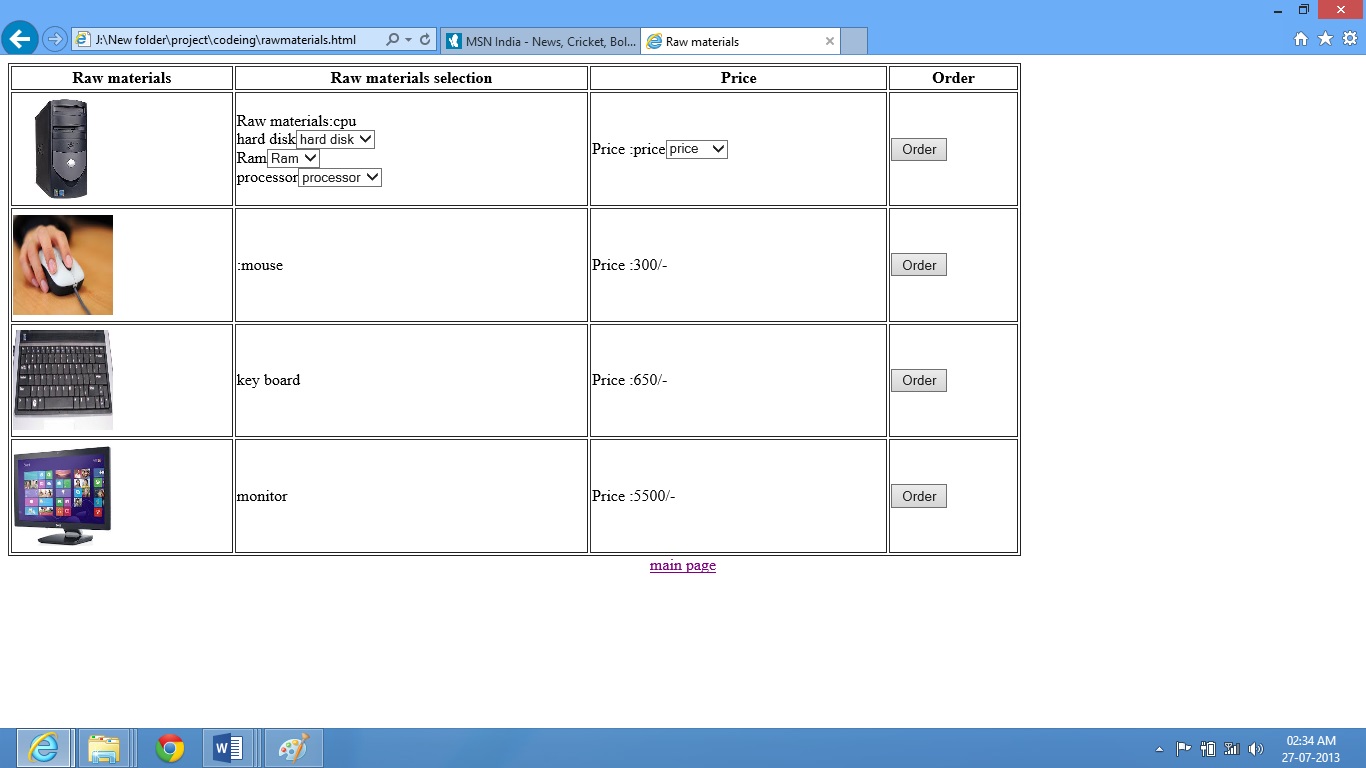
**Login:**

****

**Manager:**



**Products:**

****

**8. CONCLUSION**

This section examines the extent to which computer-based systems are organized to enhance or degrade the quality of working lifeforclerk, administrativestaf, professionals,andmanagers. Worklife merits a lot of attention for four reasons.

First, work is a major component of many people's lives. Wage income is the primary way that most people between the ages of 22 and 65 obtain money for food, housing, clothing, transportation, etc. The United States' population is about 260,000,000, and well over 110,000,000 work for a living. So, major changes in the nature of work the number of jobs, the nature of jobs, career opportunities, job content, social relationships at work, working conditions of various kinds can affect a significant segment of society.

Second, in the United States, most wage earners work thirty to sixty hours per week--a large fraction of their waking lives. And people's experiences at work, whether good or bad, can shape other aspects of their lives as well. Work pressures or work pleasures can be carried home to families. Better jobs give people some room to grow when they seek more responsible, or complex positions, while stifled careers often breed boredom and resentment in comparably motivated people. Although people vary considerably in what kinds of experiences and opportunities they want from a job, few people would be thrilled with a monotonous and socially isolated job, even if it were to pay very well.

Third, computerization has touched more people more visibly in their work than in any other kind of setting--home, schools, churches, banking, and so on. Workplaces are good places to examine how the dreams and dilemmas of computerization really work out for large numbers of people under an immense variety of social and technical conditions.

**9. BIBLOGRAPHY**

1. Database Management System

-C.J.Data

2. JAVA2 The complete Reference

-Patrick Naughton

3. Pure Java Script

-R.Allen Wyke & Charlton Ting

1. Hyper Text Markup Language
2. java.sun.com/products/**jsp/**
3. [**www.javascript.**com](http://www.javascript.com)
4. Jakarta.**apache.**org/**tomcat**
5. java.sun.com/**j2ee**
6. www.w3schools.com

**10. APPENDIX (USER MANUAL)**

* First we have to install the Apache Tomcat version 5.5.0.
* Then we have to install MS Access 2003.
* Then Create the Database.
* Then we have to design the front end code.
* After then we have to provide the Database Connectivity By using Data Sources.
* At last we have to perform Debugging.