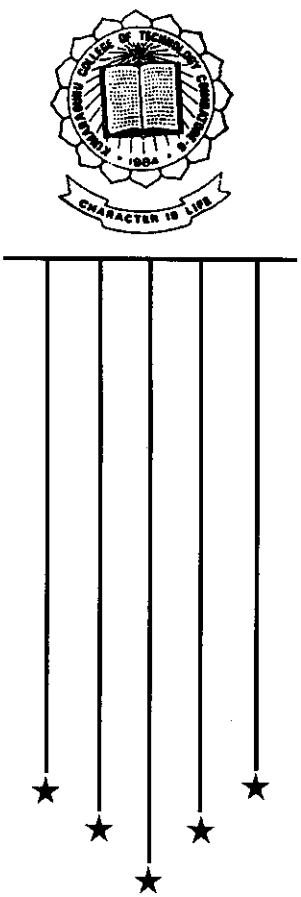
E-BAZAAR



PROJECT REPORT

Submitted By

JAYANTHI S

DEVAPRIYA D

Guided By

Mrs. D. CHANDRAKALA, M.E.,

IN PARTIAL FULFILMENT OF THE REQUIREMNTS

FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

OF BHARATHIAR UNIVERSITY

2000 - 2001

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore - 641006

BAT BHARATI
ADVANCED TECHNOLOGI
Technology for fut

Certificate

To Whomsoever It May Concern

Certified that this thesis " **E-Bazaar**" is the bona fide work of **Ms.Jayanthi Shanmugham** & **Ms.Devapriya D** who have carried out the project under our supervision, certified further that to the best of our knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate. The Documentation submitted by the group was very well compiled and completely satisfied our expectations.

Director

A. Ganesh Kumar.

9/3/2001

Project Guide

K. Kumaresh.
9/3/2001

Kumaresh K.

11/25 - 3rd Floor, "PEOPLE'S PARK", Govt. Arts College Road, Coimbatore - 641 018. INDIA
Phone : 210266 Cell : 98422 - 30030 E-mail : gkbat@eth.net
www.batindia.com

DECLARATION

DECLARATION

We, **Devapriya D.**, 9727K0133, **Jayanthi Shanmugham**, 9727K0145, students of B.E. (CSE) from Kumaraguru College of Technology (affiliated to Bharathiar University) hereby declare that the project entitled "E- BAZAAR" is our original work developed for MDC FUTURA, Coimbatore during the period 2000 –2001.This project has been done as a part of our B.E. Curriculum.

Dedicated to our Beloved Parents

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any taste 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 sincerely thank our Principal **Dr.K.K.Padmanaban**, for the patronage and innumerable facilities provided by him for the project work.

Our heartiest thanks are due to our patron, **Prof.S.Thangaswamy**, Head of the Department of Computer Science and Engineering, for his kind guidance and support.

We would like to express our sincere thanks to our internal guide, **Mrs. D. Chandrakala**, for her valuable guidance and encouragement during the course of the project.

We extend our heartiest gratitude to our Class Advisor **Ms.A.Lavanya**, for all the help she extended to us in our project work. We also thank all the faculty members and lab technicians for their kind help and support.

We express our deep sense of gratitude to our external guide **Mr. Kumaresh**, our project co-coordinator for their immense help and guidance during the course of the project work. We extend our heartiest thanks to all members of the BAT Team for their valuable suggestions and encouragement.

We express our sincere thanks to **Mr.A.Ganesh Kumar,** Chief Executive Officer, Bharath Advanced Technologies (BAT), for providing us this project.

Finally, we sincerely thank all our **friends** who have contributed in the form of ideas, constructive criticisms and encouragement for the successful completion of the project.

SYNOPSIS

SYNOPSIS

This project work entitled **"E-BAZAAR"** is a wed based application developed using Servlets for Bharath Advanced Technologies, Coimbatore.

This Online Shopping aims at providing online services and help to it's registered customers and users. There are group of products which are registered by the manufacturers and available online so that the customers can buy their wished products.

After selecting their products, the quantity is entered by the customer and the total amount is calculated. The customers can pay their amount using their credit card and hence the products will be delivered to them during a particular course of time.

CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 WORLD WIDE WEB

The Net information superhighway, Cyberspace, The web chances are that you already know more than enough about these magic buzzwords which have been splattered across the billboards, TV commercials, and magazines advertisements lately. What was once only in the domain of the technology elite has become the small talk of everyday people across the nation.

The WWW, i.e., the World Wide Web, in particular, has exploded into the scene and has promises to change the way we work, learn and play. The World Wide Web is one of the most significant inventions since the personal computer. The web is actually built upon a much more comprehensive network known as the INTERNET, without which any of the hype and hysteria surrounding the web itself would mean a thing. Without the Internet the World Wide Web would not mean a thing and Vice Versa. The world wide web, known fondly to millions as WWW, W3 or simply as "the web". The web is actually a superset of the Internet. It can be thought of as a graphical interface to the Internet, providing a revolutionary way of accessing information scattered across millions of computers around the world. Before World Wide Web, the Internet was very difficult to navigate.

THE WEB PAGES

The web pages are the most visually captivating aspect of the entire system and what we "tune in" to see. The term "page" refers to nothing to nothing more than a World Wide Web document. A web page is more like a word processing document than a printed page, especially in the way it is viewed. All the information on a page can be viewed by scrolling. In a nutshell, a page refers to all related information in a single web document.

WEB SITES

A World Wide Web site is simply a collection of interlinked web pages. The web makes it possible to link any number of items, including other pages. Because of the inherent diversity found in the web, a site might only provide links to pages pertaining to a specific subject.

HOME PAGES

Each site has a HOME PAGE, the official access point into the site. From the Home Page, you can get an overview of the site and begin exploring it. A properly designed home page is, in essence, like a hospitable host. It invites you in and tells you about all the things you can see and do during the visit. It is the most important page of any site.

WEB BROWSER

The only way you can visually navigate the web is by using a special type of software known as a browser. A good browser makes it possible to easily navigate and retrieve information from the web, visually presenting each page you connect to.

UNIFORM RESOURCE LOCATORS (URLS)

Uniform resource locators, or URLs, are the standard way of locating and retrieving information on the World Wide Web. They tell your browser what piece of information to retrieve, how to get it, and what protocol to use the process. URLs can be thought of as postal address for the web. They specify where pages, files and other piece of information are located, making it possible for browsers to find and display, or even download, that information.

WEB DESIGN

Web Design means taking advantage of the various elements supported by the World Wide Web – including text, images, audio and animation – and orchestrating them to create a Web site that embraces the opportunities actualized by Web technology. Web design is much more than taking a printed page and scanning it, tossing some HTML tags around it, and placing it on a Web server.

Rather, a successful design incorporates the hypertextual, contextual and sensory realms and combines them with effective programming and administrative and marketing strategies. Ultimately, Web deign is the creation of high-impact presence for companies and organization seeking representation on the Internet.

COMPANIES OF DESIGN

There are various components involved in Web design. Web designing incorporates, but is not limited to the components described here. Its important to note that while all these elements are part of what a web designer needs to understand, how and when they are to be applied are critical a Web designer must make.

CONTENT AND COPY WRITING

Content is the most fundamental aspect of Web design because without contents to share, the web would serve little purpose. The Web designer having examined the needs of a given client must decide how to implement the copy within the site's layout. The contents have to be refined and made consistent with the needs of the Web site while addressing the needs of the site's visitors.

DESIGN LANGUAGE

Hyper Text Markup Language (HTML) is the language of the Web. The essential component without which nothing else could exist. Contemporary web design demands a thorough knowledge and skill to use it cross browser barriers.

GRAPHIC DESIGN

Graphic designing, on the Net, has become a hot issue under challenge as being superfluous. Graphic design on the Web has become a necessity in the commercial and competitive realm. Quality Web design means using graphics, learning the programs and the tools that best create them and designing cleverly, meeting a variety of browser and bandwidth requirements.

ORGANIZATION PROFILE

1.2 ORGANISATION PROFILE

Bharath Advanced Technologies (BAT) a limb of Bharath group was launched basically as a training center. As an addition to it's experience, BAT took the franchise of MDC Systems U.S.A. for it's training. Now BAT has grown to numerous activities based on the two major divisions viz. Training and project development.

BAT has got it's own project division. As the need for e-commerce is growing rapidly BAT provide solutions catering to almost all industrial needs, ranging from hotels to lodging, net banking to remote area business management. They also provide networking solutions and live shows on the web. They are currently planning to act as an ISP and have their branches in Singapore and US.

SYSTEM ANALYSIS

2.1 PROJECT DESCRIPTION

This project work entitled "E-BAZAAR" is a web based application. This project deals with online shopping. This web site deals with transactions between the sellers and the buyers. It includes customer registration, login session, the various product details, and the transaction flow.

The different pages in this site are as follows :

- **Home**

This shall be the front page that provides information about the different pages in the site, their functions, the services they provide, and also has hyperlinks to all pages and vice versa.

- **Customer's entry**

The customers enter their user name and password to enter the web site. It is checked with the database. If match found, the control is transferred to the main page. If not, he is asked to enter the name and password again. If he is a new user, he can register into our web site.

- **New User Entry**

The new users have to register first before starting the shopping. He is asked details about his name, password, date of birth, address, city, state, e- mail id. If the informations are valid, the control is transferred to the main page.

- **Main page**

The main page provides the list of all the various products currently available for shopping. Each product on the list is a hyperlink. Once we click a product link, the detailed information about the products is displayed. For example, the auto components hyperlink leads to the page containing details about different auto components. The price and details for each product can be seen.

- **Adding products to cart**

Each product can be added to the shopping cart by clicking the select button present. The products selected will be entered into the database along with its price and default quantity (1), which can be modified later.

- **Credit Card entry**

Once the selection process is over, the customer will be sent to the credit card entry page wherein he is asked to enter his credit card number. The customer's name and credit card number is automatically stored in the database for validation.

- **Bill**

The final stage in the shopping process is the processing of the bill. The selected products along with their price and default quantity is displayed as a table. The quantity can be changed according to his wish. If he does not need a particular product from the table, the quantity can be entered as zero. On the click of a button, the total amount will be calculated and the final bill is produced.

PROGRAMMING ENVIRONMENT

HARDWARE SPECIFICATION:

- Pentium II 350 Mhz
- 64 MB RAM
- GB Hard Disk

SOFTWARE SPECIFICATION:

- Windows Millenium Version
- Java – Servlets
- CORBA
- MS - Access 2000 as backend

SELECTION OF SOFTWARE

2.2 SELECTION OF SOFTWARE

CORBA is a specification for an emerging technology known as distributed object management (DOM). DOM technology provides a higher-level, object-oriented interface on top of the basic distributed computing services. CORBA has emerged as the leading standard among DOM solutions. CORBA defines a standard framework from which an information system implementer or software developer can easily and quickly integrate network-resident software modules and application to create new, more-powerful applications. It combines object technology with a client server model to provide a uniform view of an enterprise's computing system – everything on the network is an object.

The highest-level specification is referred to as the object management architecture (OMA), which addresses four architectural elements. The term CORBA is often used to refer to the Object Request Broker (ORB) itself, as well as to the entire OMG architecture.

The role of the ORB is to route requests among the other architecture components. CORBA services, CORBA facilities, and CORBA domains are also defined as part of the specifications. CORBA services provide some basic system-level services, such as Naming, Persistence, and Event Notification. CORBA facilities are a set of higher-level functions that cover a broad range of generically

applicable facilities in areas such as user interface and information management. CORBA domains are specific to particular application domains, such as manufacturing, financing, and telecommunications. Finally, application objects provide the new business capabilities that are created by the system implementers.

The key to integrating application objects is the specification of standard interfaces using the interface definition language (IDL). Once all applications and data have an IDL- compliant interface, communication is independent of physical location, platform type, networking protocol, and programming language. An information system is created by using CORBA to mediate the flow of control and information among these software objects.

CORBA CONCEPTS:

- Object- oriented model
- Open distributed computing environments
- Component integration and reuse

CORBA provides unified access to applications, independent of the location of each application on the network. It provides uniform access to services, uniform discovery of resources and object names, uniform error handling methods and uniform security policies.

The broker plays two key roles. First, it provides common services, including basic messaging and communication between client and server, directory services, meta-data description, security services, and location transparency. It also insulates the application from the specifics of the system configuration, such as hardware platforms and operating systems, network protocols, and implementation languages.

CORBA is based on a peer-to-peer communications model and supports both synchronous and a limited version of asynchronous communications. The underlying interprocess communications in CORBA is handled through the general inter-ORB protocol (GIOP). The GIOP was specified to allow interoperability among ORB implementations. It assumes an underlying connection-oriented transport layer.

The mapping of GIOP to TCP/IP is referred to as the Internet inter-ORB protocol (IIOP). The ORB itself handles the communications routing of a request, as well as any ancillary services required to perform the request.

SOFTWARE DESCRIPTION

3. SOFTWARE DESCRIPTION

3.1 JAVA – SERVLETS

The power of servlets is

1. Portability

Because servlets are written in java, they are highly portable across operating systems and across server implementations. We can develop a servlet on a Windows NT machine running the java web server.

2. Power

Servlets can harness the full power of the core java-networking, multithreading, image manipulation, data compression, remote method invocation (RMI), and CORBA connectivity and object serialization.

3. Efficiency

Servlet invocation is highly efficient. Once a servlet is loaded, it remains in the server's memory as a single object instance. Thereafter, the server invokes the servlet to handle a request using a simple, lightweight method invocation. Servlets are highly

scalable since multiple, concurrent requests are handled by separate threads.

4. Safety

Servlet's can handle the error safely, due to the java's exception-handling mechanism. If a servlet divides by zero or performs other legal operations, it throes an exception that can be safely caught and handled by the server. A server can protect itself from servlets thro the use of java security manager.

5. Integration

Servlets are tightly integrated with the server. This integration allows the servlet to co-operate with the server. Servlet can use the server to translate file paths, perform logging, check authorization and even add users to the server's user database.

3.2 CORBA

INTRODUCTION

In 3-Tier architecture, component development plays a vital role. Component architecture enables the programmer to develop components which can offer services to clients irrespective of platforms and development tools.

CORBA is a component architecture specification, which emphasis how two components can interact with each other. Common Object Request Broker Architecture enables components developed in C++ , Java or any other language to offer services irrespective of languages and platforms. CORBA components are packaged as binary components. Any remote client can avail the services from these components with method invocation. The CORBA component publishes its service to the outside world with interfaces. The interface is the binding contract between clients and components.

An interface is the declaration of the services offered by a CORBA component. A CORBA component can offer services. These services are published by the component using interfaces to enable the clients to avail services. These interfaces are purely declarative which don't have any definition. The interfaces will be written using the neutral declarative language IDL (Interface Definition Language). Once after the interface declaration the

component can be implemented in any language (C, C++, Java, Ada). The client needs to know only the services published by the interface. There is no necessity for the client to know about the server implementation code, otherwise the server code is completely transparent to the client.

IDL can be used to specify a component's attributes, the parent classes it inherits from, the exceptions it raises, the typed events it emits and the methods the interface supports- including the input, output argument and their corresponding data types.

Objects can discover each other and invoke services with the CORBA ORB.

BENEFITS OF CORBA ORB

1.STATIC AND DYNAMIC INVOCATIONS

In CORBA ORB communications the method invocations can be informed during compile time or the client can discover the methods during the run-time. This enables the programmers to have both strong-type checking and late binding functionality.

2.HIGH LEVEL LANGUAGE BINDINGS

The CORBA ORB enables the developer to invoke methods form server objects irrespective of language implementation. That means there is no need for the client object to know about the high level language that is used to implement the server object. This is because the services offered by the server object are published in IDL, which is independent of any high level language.

3.SELF-DESCRIBING SYSTEM

The ORB supports *Interface Repository,* which offers run-time information describing the functions offered by the server. This interface provides run-time meta-data information about all services offered by the CORBA components. The clients can use the Interface Repository dynamically to discover the methods available. The Interface Repository is populated after compiling the IDL definitions of corba components.

4.LOCAL/REMOTE TRANSPARENCY

CORBA offers Local and Remote Transparency. The CORBA ORB can be interconnected with other ORB'S in the network and offer services or it can run in the stand-alone mode. The IIOP (Internet Inter ORB Protocol) enbles the

communication between ORB's across the network irrespective of platforms. This makes any CORBA component to be identified in the internet environment. Another major advantage of CORBA ORB is there is no need to worry about the dependencies, OS internals etc., from the developer's point of view. The developer need to know only what service he want's to invoke.

5.BUILT-IN SECURITY

The context information provided in the messages by the ORB enables to handle the security and transactions across machine and ORB boundaries.

THE CLIENT SIDE CORBA

1.CLIENT IDL STUBS

Stubs are pieces of pre-compiled code which provide static interfaces to object services. This enables clients to invoke any methods on server objects. The stubs are termed as a local proxy for the remote object. The IDL compiler is used to compile the IDL file, which will create the stubs. The stubs encode and decodes the methods and it's parameters into the standard format that can be sent to the server. This is termed as marshalling.

2.DYNAMIC INVOCATION INTERFACE

Enables to discover the methods to be invoked at runtime. CORBA API's can be used to discover the methods form the meta-data repository, issue the remote call and get back the results.

3.INTERFACE REPOSITORY

IR API's allows us to modify the details of the components registered such as it's interfaces, the methods they support, and the parameters they require. The IR is the runtime distributed database that contains binary versions of the IDL –defined interfaces. The API's allows components to dynamically access, store and update metadata information. This enables the entire ORB as a self describing system.

SERVER SIDE ORB

1.SKELETONS

Provides static interfaces to each service exported by the server. The IDL pre-compiler creates these skeletons.

2.DYNAMIC SKELETON INTERFACE(DSI)

DSI enables any client components, which do not have the IDL stubs to avail the service from the Remote object. The Dynamic Skeleton does this by identifying the arguments in the incoming message to understand the target object for which the request is made. Dynamic Skeletons are very useful for implementing generic bridges between ORB's. The DSI is the server equivalent of DII. It has the ability to receive either static or dynamic invocations.

3.OBJECT ADAPTER

Accepts the requests for services on behalf of the server objects from the remote clients. This offers the run-time environment for instantiating server objects, passing requests to them and assigning them object IDI's. The Object Adapter registers the classes it supports and their run-time instances with the implementation Repository. CORBA specification emphasizes that each ORB must support a standard adapter called the Basic Object Adapter (BOA). Servers may support more than one object adapter.

4.IMPLEMENTATION REPOSITORY

Provides run-time repository of information about the classes a server supports, the objects that are instantiated, and their ID's. It serves as a common place to store additional information associated with the implementation of ORB's.

5.ORB INTERFACE

Contains API's to local services that can be used by the application.

3.3 MS ACCESS 2000

In other database-management programs, the term database is sometimes used to refer to tables that hold data. An Access database consists of the tables that hold the data and all the related objects, such as queries, forms and reports that are used to manage the data. When you open the database, Access displays the database windows, which is sometimes called that make up the database.

We can create tables in the design view or the typical wizard. We also have the capability to create forms. Forms let us control how data is displayed on the screen. We also have an option for printing data using the report option.

The powerful features of Access are:

- Macros that let you automate and speed up your work; they are also used when you develop applications. A macro is a list of actions. Access performs all the actions in the list when you run the macro. Macro saves time for Access users.
- Modules that let to you write programs in Visual Basic to develop advanced applications.

We have a lot of utilities and special techniques present in Access.

They are listed below:

- Using Access utilities to mange databases and their objects
- Using hyperlink data types
- Creating web pages
- Creating indexes based on single or multiple fields
- Working with both embedded and linked OLE objects in forms and queries
- Attaching a table from another database application so it can be used by Access and other application simultaneously
- Customizing the Access working environment with the Options dialog box.

To sum up Access being with database utilities that let you compact, convert, encrypt and repair databases , and object utilities that let you rename , delete, cut, copy and paste.

DEVELOPMENT PHASE

4.1 DATABASE DESIGN

Tables

1. Prods
2. List
3. Info
4. detail
5. Credit

Structure of customer entry table

S.No.	Name of the Field	Data Type	Width	Description
1.	User Name	Text	20	Users name
2.	Passwd	Text	8	Password
3.	Addr	Text	50	Address
4.	DOB	Date		Date of birth
5.	State	Text	15	
6.	Coun	Text	15	Country
7.	Email	Text	15	Email id

Structure of products list table

S.No.	Name of the Field	Data Type	Width	Description
1.	Prodname	Text	20	Product name
2.	Price	Number	10	Price
3.	Qty	Number	10	Total Quantity

Structure of selected products table

S.No.	Name of the Field	Data Type	Width	Description
1.	Prodname	Text	20	Product name
2.	Price	Number	10	Price
3.	Qty	Number	10	Total Quantity

Structure of info table

S.No.	Name of the Field	Data Type	Width	Description
1.	Name	Text	20	Customer's Name
2.	Cno	Number	10	Credit card number
3.	Amount	Number	10	Total Amount

Structure of credit information table

S.No.	Name of the Field	Data Type	Width	Description
1.	Name	Text	20	Customer Name
2.	Cno	Number	10	Credit card Number

4.2 MODULE DESIGN

Creation of Database

The database design has been implemented in MS Access 2000 with five tables.

ODBC Connectivity

The Open Database Connectivity is a programming interface developed by Microsoft that allows the database to be interfaced with any other client.

Screens

This system provides numerous highly controlled user interfaces built with the help of powerful tools. Some of the user interface and their functions are listed below.

Customer's Registration Form
Customer's Login Form
Products Description Forms
Transaction Forms
Credit Card Entry Form
Bill Processing Form

IMPLEMENTATION

5. IMPLEMENTATION

Count.idl

This program provides the interface between Java and Corba. The language used is Interface Definition Language. It contains the definitions of all the functions used in the count Impl program. It can be compiled using idl2java compiler (to convert into java) or idl2c++ compiler (to convert it to C++). On compilation, nine java programs are generated. They are,

1. countHolder.java
2. countHelper.java
3. countOperations.java
4. _st_count.java
5. _sk_count.java
6. _tie_count.java
7. _countImplBase.java
8. count.java
9. _example_count.java

countImpl.java

Customer's login:

The check function of this program is used to check if the user is already registered. The name and password are passed from the servlet, and they are checked with the database. After checking, a boolean value is returned to the servlet.

Customer's registration:

The new user function is used for their purpose. Their name, address, password, date of birth, email id, country and state informations are passed as parameters and entered into the table.

Selection of Products:

The order function is used to enter the selected products into the database. The product name, customer name and price are entered into the table.

Credit Card Entry:

The credit function is used to enter the customer name and credit card number into the credit table. The final 1 function lists the selected products along with their default quantity and its price in a table. The customer can change the quantity.

Bill Transactions:

The update function is used to calculate the total amount. It also inserts the customer name, credit card number and total amount in the info table.

SYSTEM TESTING

6. 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 all is correct and an opportunity to show the user that the system works.

When we have tested each page individually, using the test data designed by us and have verified that these programs link together the ways specified in the program suite specification, the complete system and its environment was 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 DEVELOPMENT

7. SCOPE FOR FUTURE ENHANCEMENT

This project entitled "E- BAZAAR" can be further furnished with **e-mail**, **auction** and **chat.**

The e-mail part of the project, **mail,** can be added to allow customers and other end- users to place orders in advance.

The chat part of the project, **prod chat**, can be added to allow customers to discuss about the quality of the products. The auction part of the project, **auction** can be added to allow end users to put up their products for auction sale.

CONCLUSION

8. CONCLUSION

The "E-BAZAAR" has been developed to take care of the customers and extend help to them. This has removed the difficulty of the customers personnel visit to the shops and brought the shops to their home through Internet.

The design and development of such a real time project concerned with web design and publishing has been really a wonderful experience for two of us. We discovered new Internet facts in the world of Internet. Increased our degree of skill of programming gained immense knowledge on web designing and got well exposed to really a powerful architecture CORBA.

Thus this site plays a major role in a customer's life and provides information to all those who visit this site.

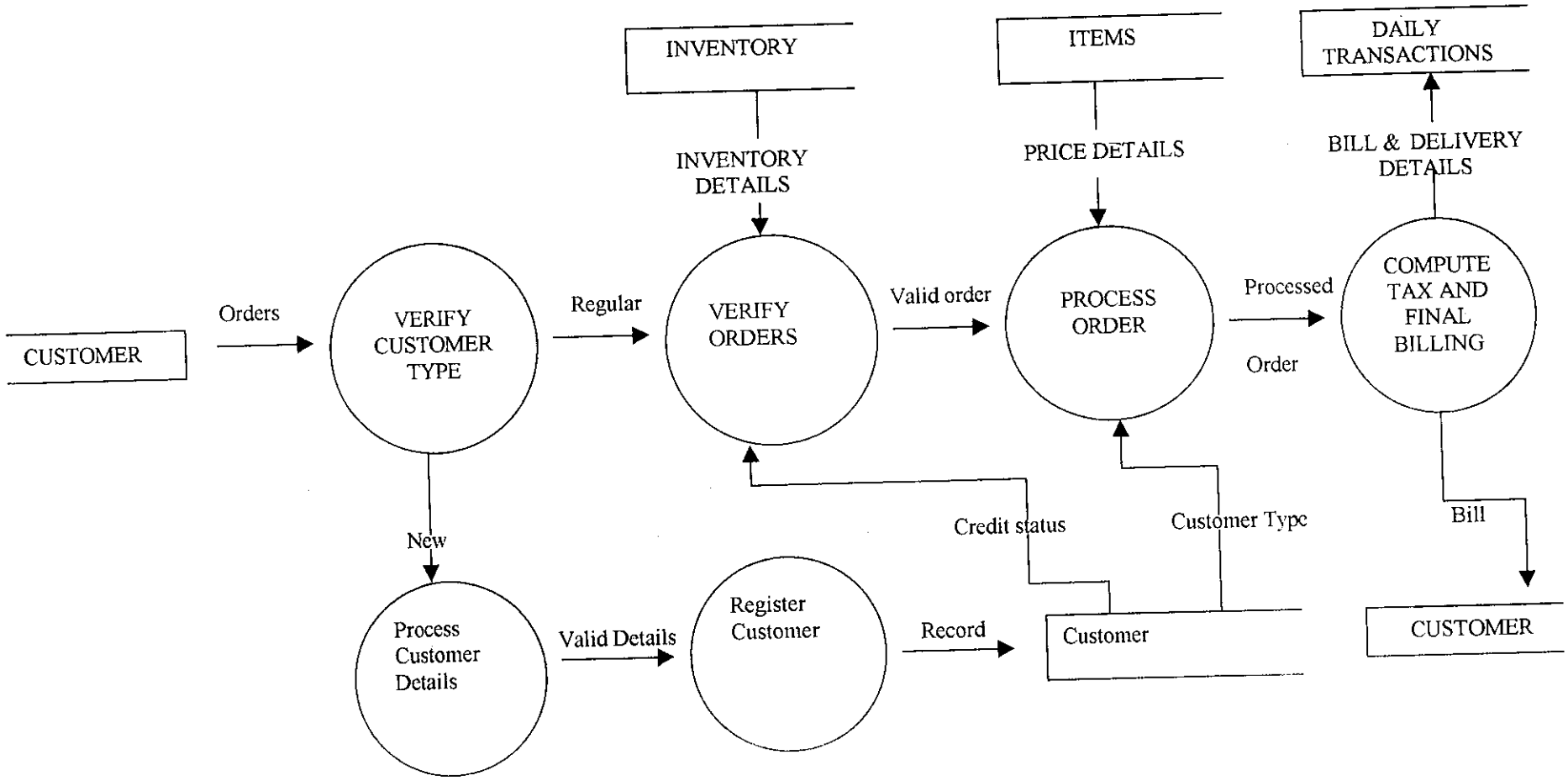
BIBLIOGRAPHY

BIBLIOGRAPHY

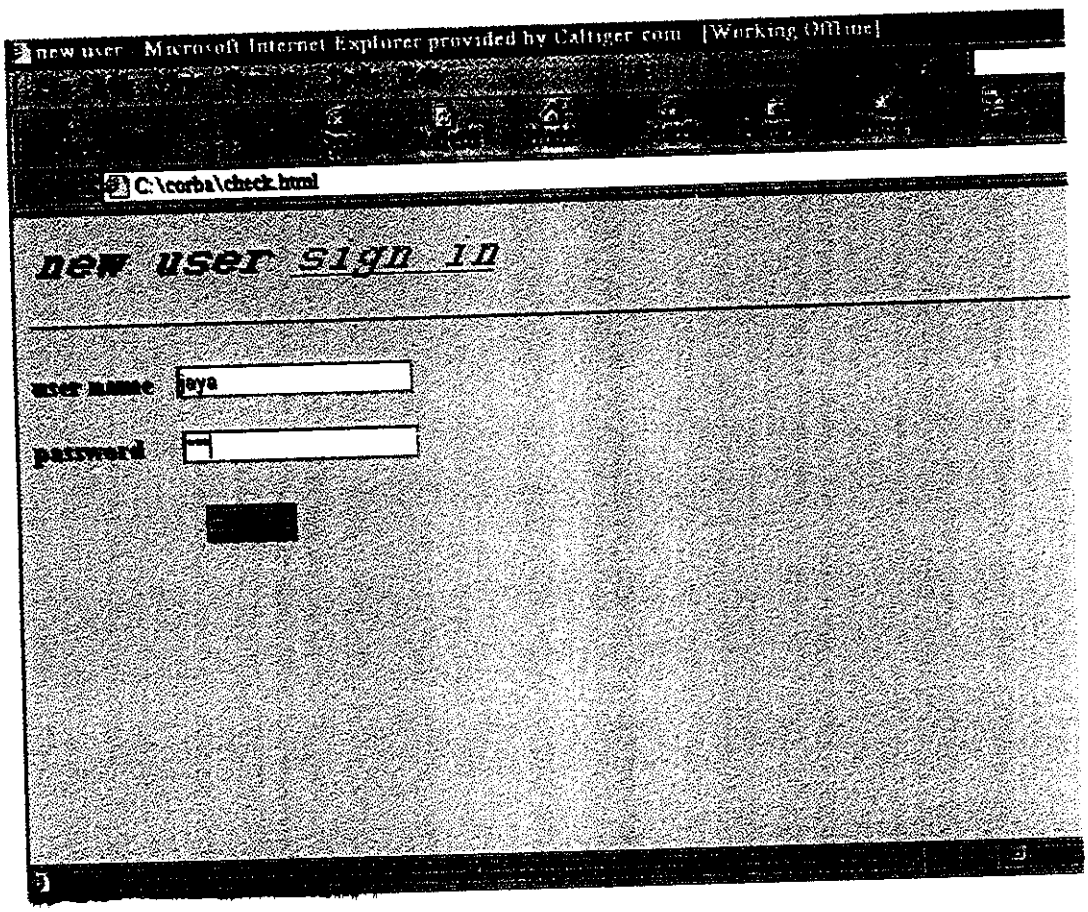
- Robert. A. Orfali., Java and CORBA.
- James Goodwill., Basic Servlet Programming.
- Thomas. J. Mowbray., Inside CORBA.

APPENDIX

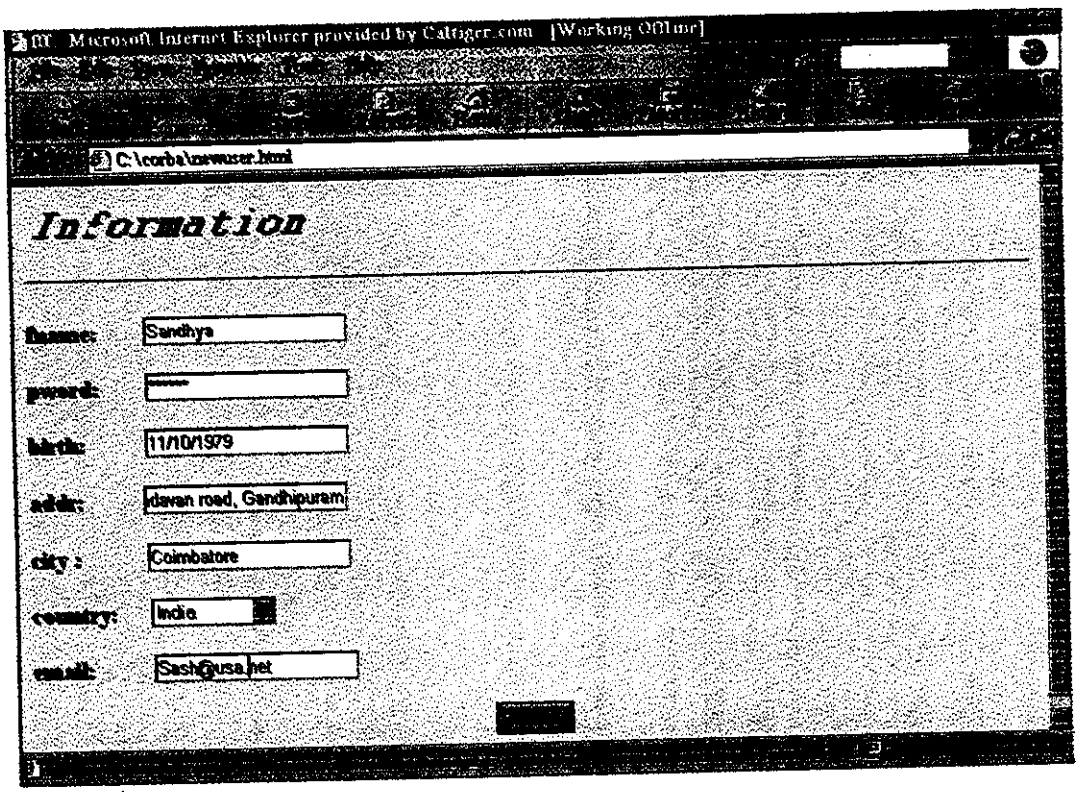
Data Flow Diagram



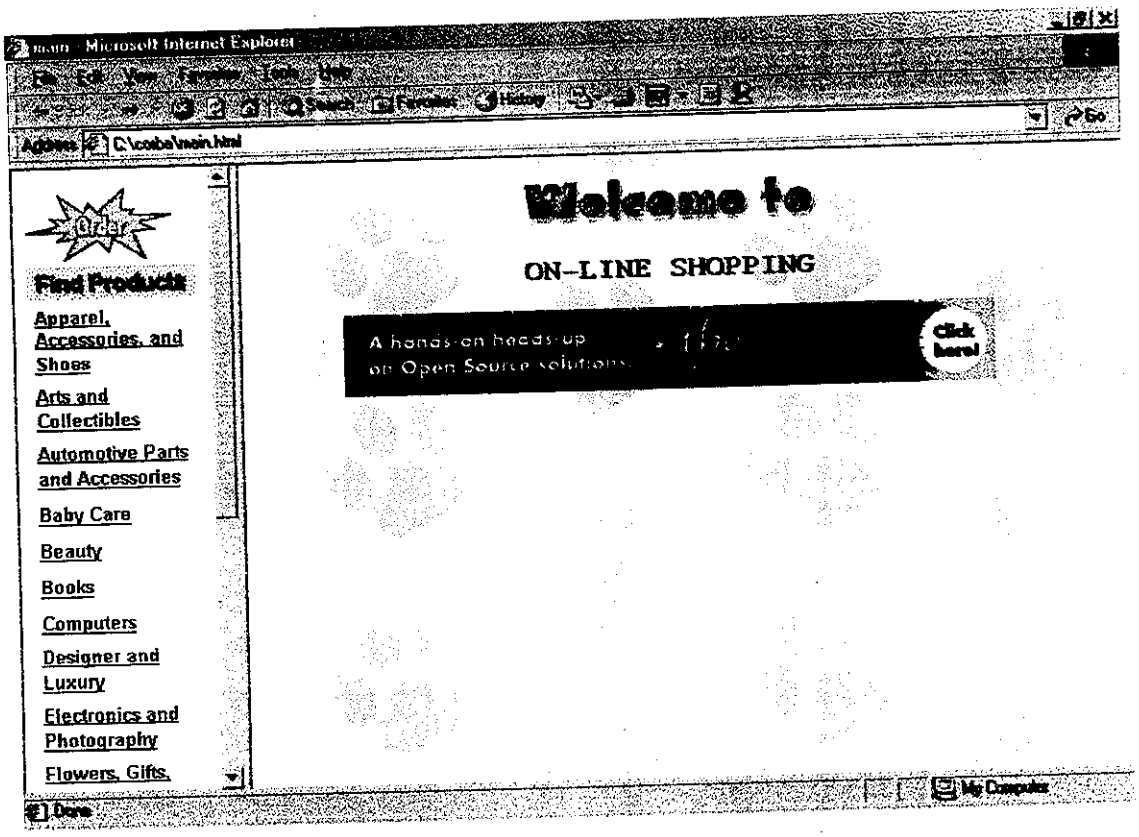
Customer's Login



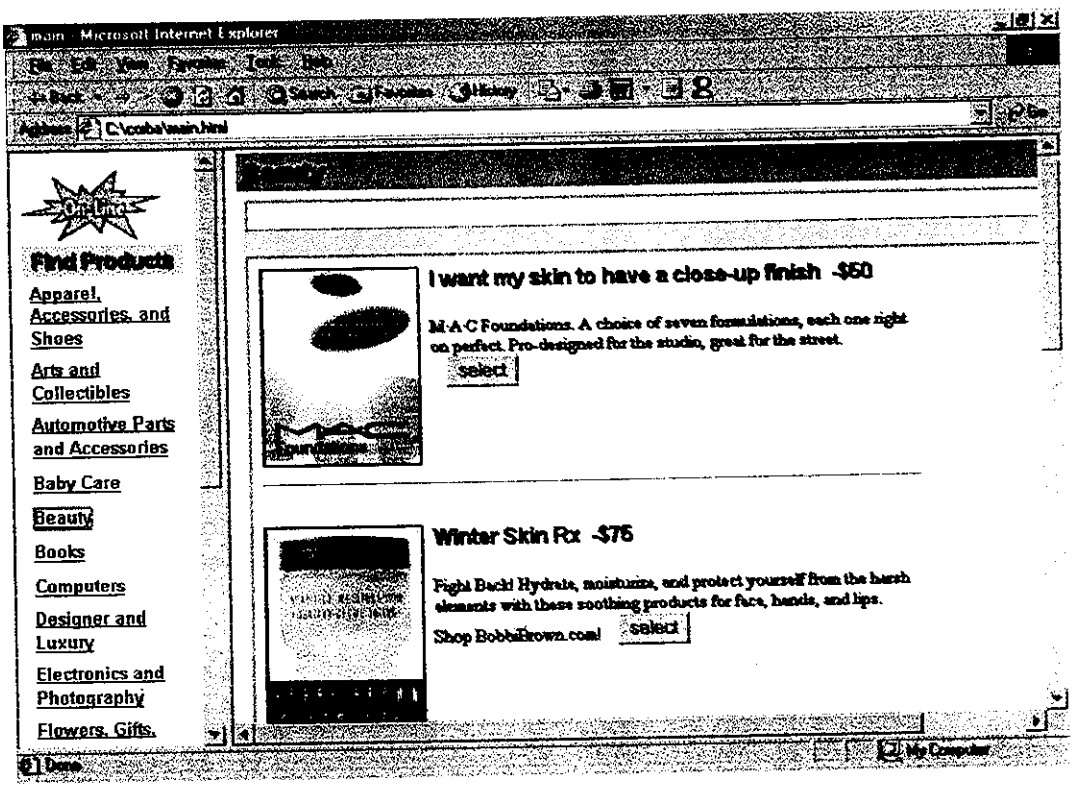
Customer's Registration



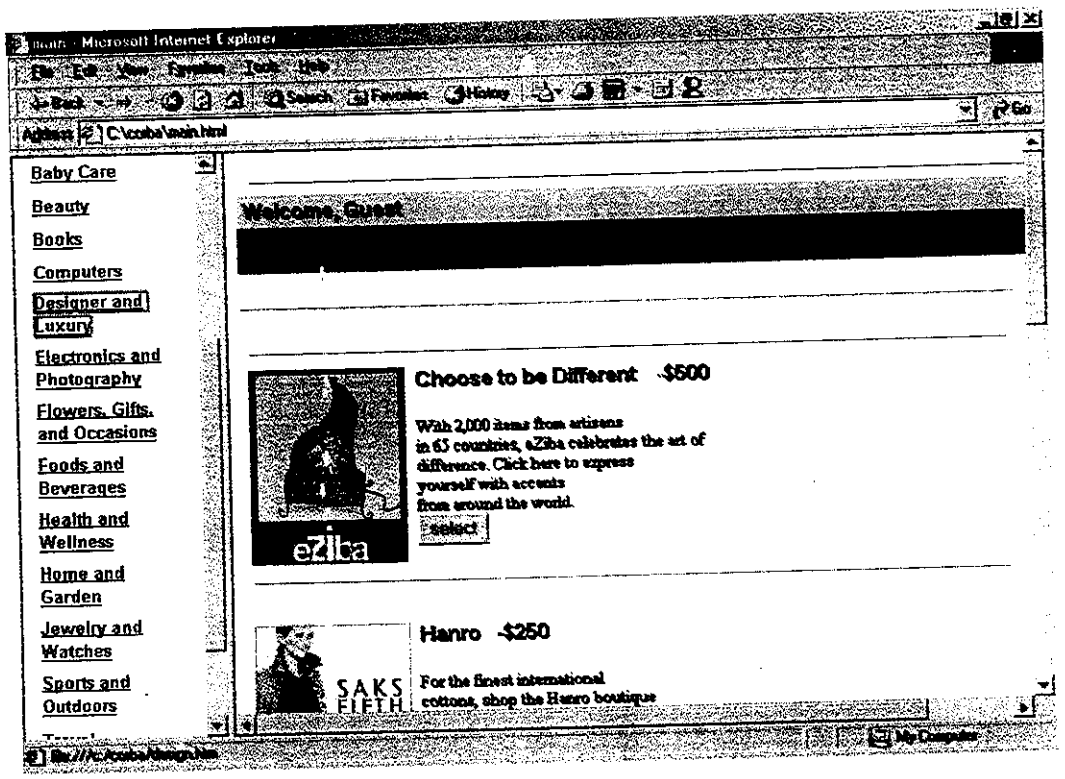
Main page


Microsoft Internet Explorer
Address C:\coba\main.html
Order
Find Products
Apparel, Accessories, and Shoes
Arts and Collectibles
Automotive Parts and Accessories
Baby Care
Beauty
Books
Computers
Designer and Luxury
Electronics and Photography
Flowers, Gifts,
Welcome to
ON-LINE SHOPPING
Click here!
Done
My Computer

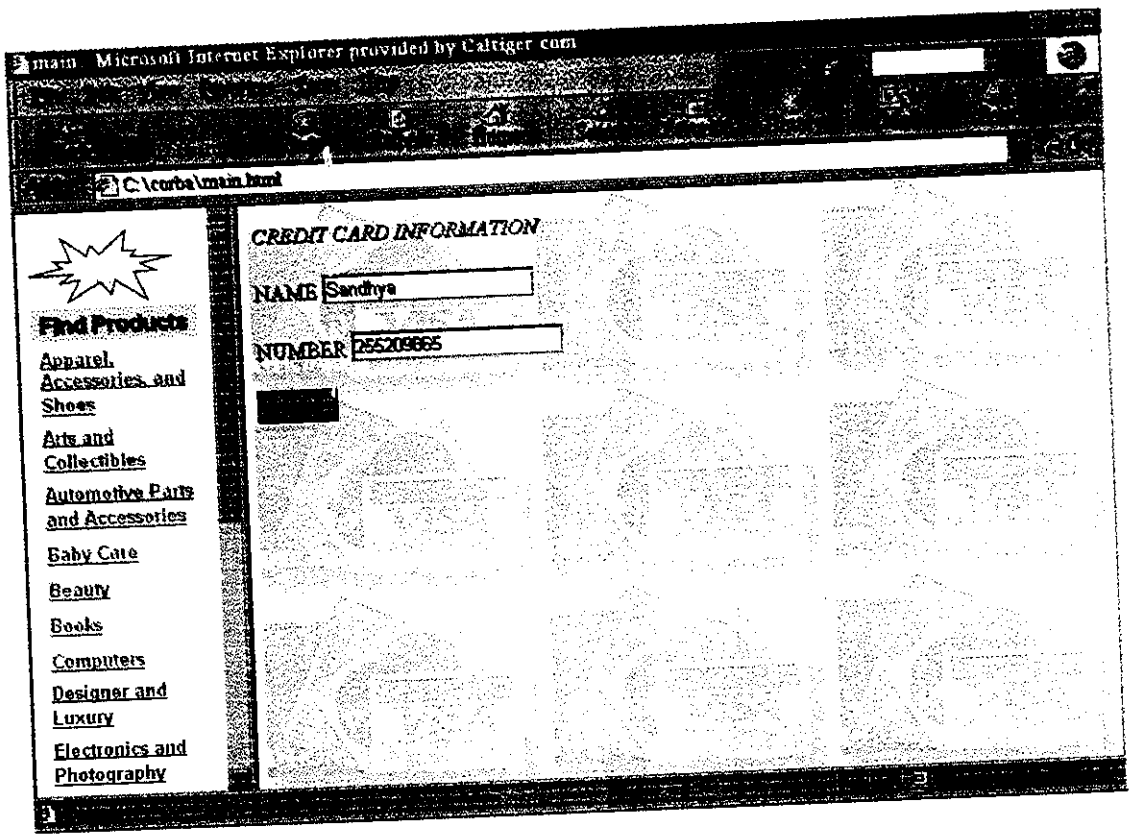
Product Detail's
Beauty



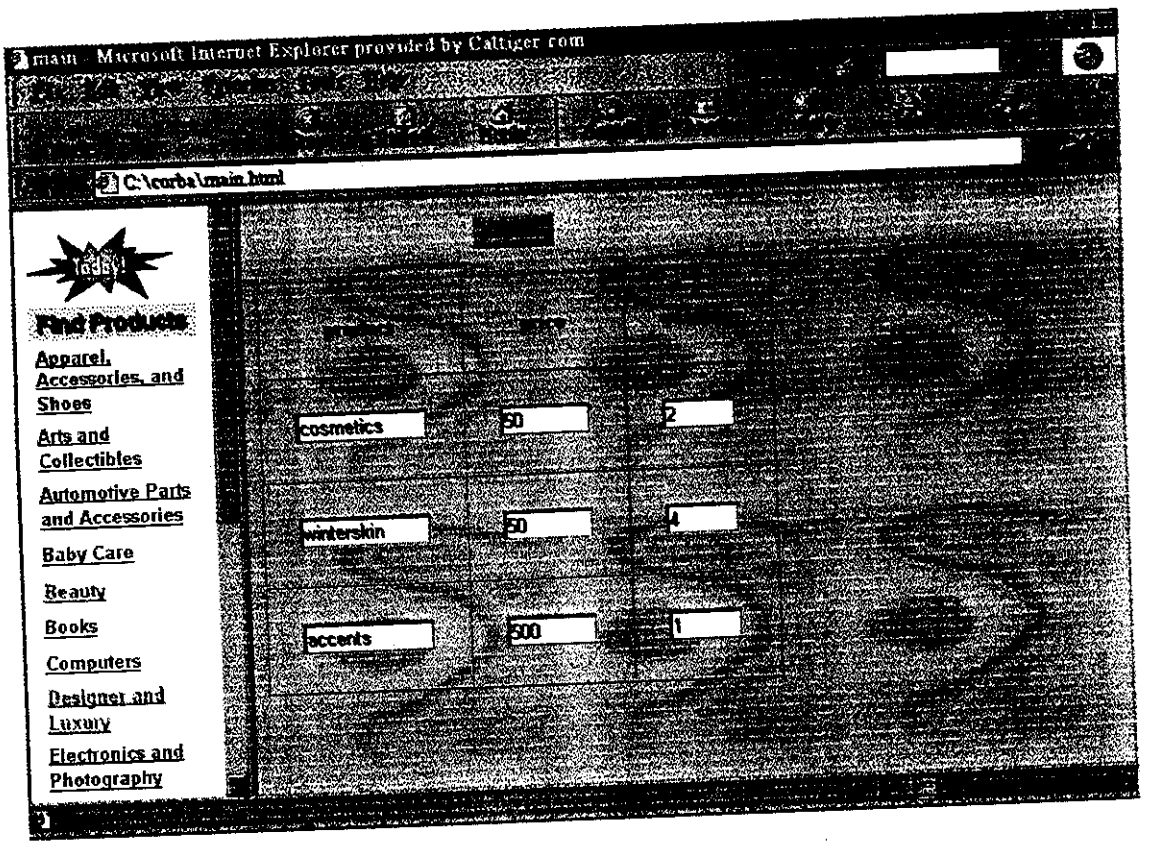
Product Detail's Designer and Luxury



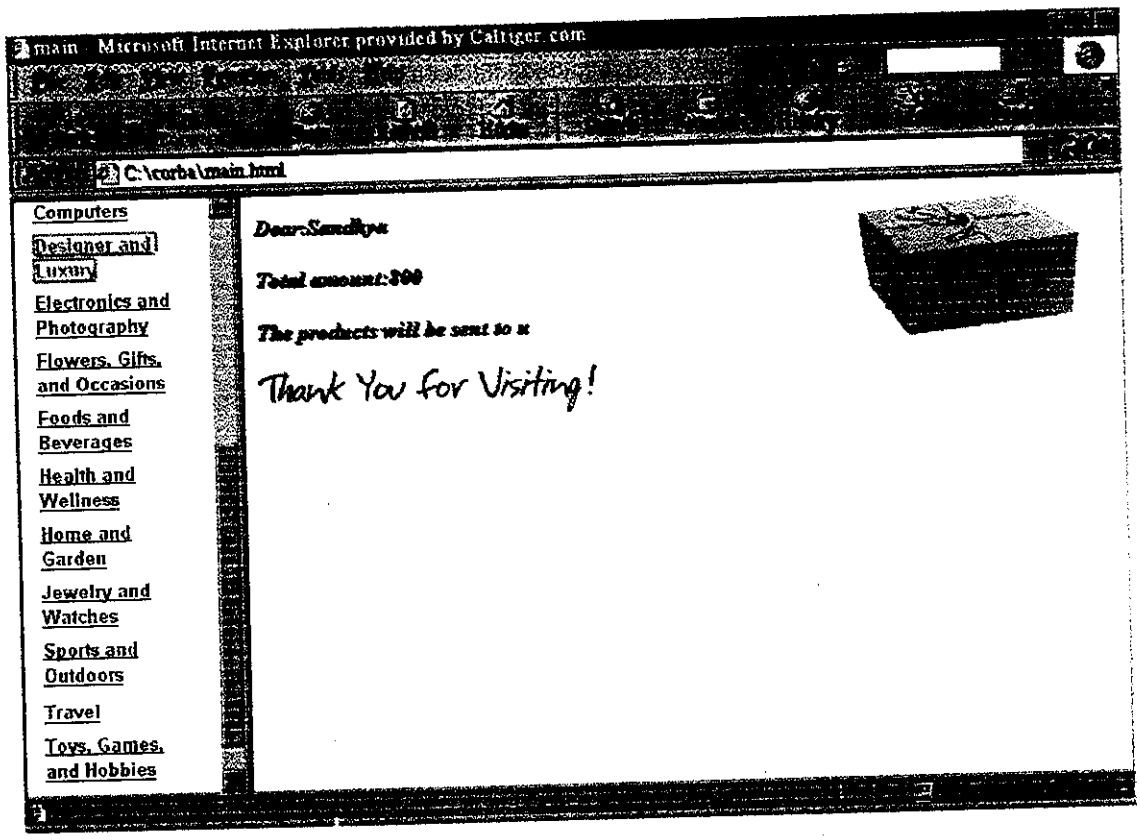
Credit Card Entry



Ordering Quantitites



Transaction of Bill



Count.idl

```
interface count
{
boolean check(in string a , in string b);
boolean insert(in string p,in string q,in string r,in string s,in string t,in string u,in
string v);
boolean order(in string q,in string w,in string name);
string bill(in string x,in string c,in string name);
string name(in string a,in string name);
string pass(in string a,in string name);
string final1(in string name);
string update(in string name,in string m,in string n);
void add(in string m,in string n);
};
```

countimpl.java:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import java.lang.String.*;
class countImpl extends _countImplBase
{
public countImpl(java.lang.String name)
  {
  super(name);
  }
public countImpl()
  {
  super();
  }
public boolean insert(String a,String b,String c,String d,String e,String f,String g)
{
 boolean as=true;
 try
 {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection co=DriverManager.getConnection("jdbc:odbc:db1");
    PreparedStatement pstmt;
    pstmt=co.prepareStatement("insert into priya1
    (name,pass,dob,addr,city,coun,email) values(?,?,?,?,?,?,?)");
    pstmt.setString(1,a);
    pstmt.setString(2,b);
    pstmt.setString(3,c);
    pstmt.setString(4,d);
    pstmt.setString(5,e);
    pstmt.setString(6,f);
    pstmt.setString(7,g);
    pstmt.executeUpdate();
    Statement se=co.createStatement();
```

```
   ResultSet rs1=se.executeQuery("delete * from table1");
}
 catch(Exception e1)
 System.out.println(e1);
 return as;
}
public boolean order(String a,String b,String name)
{
  boolean asf=true;
  try
  {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection co=DriverManager.getConnection("jdbc:odbc:db1");
    PreparedStatement pstmt;
    pstmt=co.prepareStatement("insert into table1(prod,price,name) values (?,?,?)");
    pstmt.setString(1,a);
    pstmt.setString(2,b);
    pstmt.setString(3,name);
    pstmt.executeUpdate();
  }
  catch(Exception eq)
  System.out.println(eq);
  return asf;
}
 public boolean check(String a,String b)
 {
  boolean asd=false;
  try
  {
     String u="",y="";
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
     Connection c=DriverManager.getConnection("jdbc:odbc:db1");
     Statement s=c.createStatement();
     ResultSet rs=s.executeQuery("select * from priya1");
     while(rs.next())
     {
     u=rs.getString(1);
     if(u.equals(a))
     {
                y=rs.getString(2);
                if(y.equals(b))
                    {
                        asd=true;
                        ResultSet rs1=s.executeQuery("delete * from table1");
                    }
                else
                    asd=false;
    }}
  catch(Exception e)
       System.out.println(e);
       return asd;
 }
  public String name(String a,String name)
  {
    String h="";
    int j=0,k;
```

```
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   Connection c=DriverManager.getConnection("jdbc:odbc:db1");
   Statement s1=c.createStatement();
   PreparedStatement pst;
   pst=c.prepareStatement("select * from table1 where name=?");
   pst.setString(1,name);
   ResultSet rs=pst.executeQuery();
   j=Integer.parseInt(a);
   for(k=0;k<=j;k++)
   rs.next();
  h=rs.getString(1);
  catch(Exception e)
  System.out.println(e);
  return h;
}
 public String pass(String a,String name)
 {
   String h="";
   int j=0,k;
 try
 {
 Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection c=DriverManager.getConnection("jdbc:odbc:db1");
    Statement s1=c.createStatement();
    PreparedStatement pst;
    pst=c.prepareStatement("select * from table1 where name=?");
    pst.setString(1,name);
   ResultSet rs=pst.executeQuery();
    j=Integer.parseInt(a);
    for(k=0;k<=j;k++)
    {
    rs.next();}
    h=rs.getString(2);
 }
 catch(Exception e)
 System.out.println(e);
 return h;
 }
 public String bill(String a,String b,String name)
 {
     int t=0;
     String hj="";
   try
    {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
     Connection c=DriverManager.getConnection("jdbc:odbc:db1");
     Statement s1=c.createStatement();
     PreparedStatement pst;
     pst=c.prepareStatement("insert into credit(name,cno) values (?,?)");
     pst.setString(1,a);
     pst.setString(2,b);
     pst.executeUpdate();

      PreparedStatement pstmt;
```

```
    pstmt=c.prepareStatement("select * from table1 where name=?");
    pstmt.setString(1,a);
    ResultSet rs=pstmt.executeQuery();
    while(rs.next())
    t++;
   hj=java.lang.String.valueOf(t);
}
    catch(Exception ew)
    System.out.println(ew);
    return hj;
}
public String final1(String na)
{
   int h=0;
   String  h1="";
   try
   {
     Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
     Connection co=DriverManager.getConnection("jdbc:odbc:db1");
     Statement s1=co.createStatement();
     PreparedStatement pst;
     pst=co.prepareStatement("select * from table1 where name=?");
     pst.setString(1,na);
     ResultSet rs=pst.executeQuery();
     while(rs.next())
     h++;
     h1=java.lang.String.valueOf(h);
 }
 catch(Exception e1)
 System.out.println(e1);
 return h1;
 }
 public String update(String name,String u,String q)
 {
   String tot1="";
   int tot=0;
   try
     {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      Connection co=DriverManager.getConnection("jdbc:odbc:db1");
      Statement s1=co.createStatement();
      PreparedStatement pstmt;
      ResultSet rs=s1.executeQuery("select * from table1");
      int h=0;
      String v=" ";
      int w=0,x=0,a=0,b=0,j=1,u1=0;
      ResultSet rs1=s1.executeQuery("select * from prods");
      while(rs1.next())
        {
          v=rs1.getString(1);
          if(u.equals(v))
          {
            u1=Integer.parseInt(q);
            b=rs1.getInt(3);
            x=rs1.getInt(2);
            b=b-u1;
```

```
            tot=tot+x*u1;
            pstmt=co.prepareStatement("update prods set qty="+b+" where
            prod='"+u+"'");
            pstmt.executeUpdate();
            j++;
           }
      }
    }
catch(Exception e1)
System.out.println(e1);
tot1=java.lang.String.valueOf(tot);
return tot1;
}
public void add(String name,String name2)
{
     try
     {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      Connection co=DriverManager.getConnection("jdbc:odbc:db1");
      Statement s1=co.createStatement();
      PreparedStatement pstmt;
       String n=" ";
       int f=0;
       ResultSet rs3=s1.executeQuery("select * from credit");
       while(rs3.next())
       {
          n=rs3.getString(1);
          if(n.equals(name))
          {
            f=rs3.getInt(2);
          }
       }
      int total=0;
      total=Integer.parseInt(name2);
      pstmt=co.prepareStatement("insert into info(name,cno,amt) values(?,?,?)");
      pstmt.setString(1,name);
      pstmt.setInt(2,f);
      pstmt.setInt(3,total);
      pstmt.executeUpdate();
      }
      catch(Exception ew)
      {
          System.out.println(ew);
      }
 }
```