# BOOK WORLD

# An Online Book Store

**Project Report**

Submitted By

Minal G. Shah
C.G.Raj Babu
P.L.Subhu

Guided By
M . Jayshree M.E.

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF ENGINEERING**

**Computer Science And Engineering**

Of the Bharthiar University

***Department of Computer Science And Engineering***

***Kumaraguru College of Technology***

***Coimbatore - 641 006***

# KUMARAGURU COLLEGE OF TECHNOLOGY

**COIMBATORE 641 006**

Department of Computer Science And Engineering

**Certificate**

This is to certify that the report entitled

**"Book World"**

has been submitted by

Ms./Mr...MINAL...G...SHAH.,..C.G..RAJBABU...,.P.L..SUBHU............
in partial fulfillment for the award of the Degree of Bachelor of Engineering in Computer Science And Engineering, Branch of Bharthiar University, Coimbatore 641 046 during the academic year 1997-2001.

[illegible] 10/3/01

Guide

[illegible] 10/3/01

Head of the department

Certified that the candidates with the University Registration number

1. 9727K0154
2. 9727K0164
3. 9727K0184

were examined in the project voice-voce held on 12 MARCH 2001.

.........................

Internal Examiner

.........................

External Examiner

# Declaration

We, Minal.G.Shah, Raj Babu.C.G, Subhu.P.L hereby declare that this project work entitled **'Online Shopping'** submitted to Kumaraguru College of Technology, Coimbatore (Affiliated to Bharathiar University) is a record of original work done by us under the supervision and guidance of Ms.Jayashree M.E., Department of Computer Science and Engineering.

| **Name of the candidate** | **Register Number** | **Signature of the candidate** |
|---|---|---|
| Minal.G.Shah | 9727 k0 154 | [illegible] |
| Raj Babu.C.G | 9727 k0 164 | C.G. Raj Babu |
| Subhu.P.L | 9727 k0 184 | [illegible] |

**countersigned:**

Staff in Charge:

[illegible] 10/3/01

Ms.Jayashree M.E.,
Lecturer
Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore – 641 006

Place : Coimbatore
Date : 10-3-01.

# Acknowledgement

## ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Dr.K.K.Padmanabhan,B.Sc.(Engg),M.Tech,Ph.D., our esteemed Principal, Kumaraguru College of Technology for giving us the support and encourgament that we needed for completing the project successfully.

We feel thankful to Prof.Thangaswamy,B.E.(Hons),Ph.D., The Head of the Department of Computer of Science And Engineering without whose motivation we would have failed to embark on a project of this magnitude.

We are greatly obliged to our Project Guide, Ms.Jayshree,M.E. She has been the most helpful whenever we had hitches during the makings of the project. Her guidance gave us all the confidence we needed while improvising the project.

We are extremely grateful to our Class Advisor Ms.A.Lavanya,B.E. who has always been there for us. Her support has always backed us in all our endeavors.

We are appreciative of all the help and support shown by our friends, staff of the department and the technicians, enabling us to finish the project to our fullest satisfaction.

# Contents

## CONTENTS

# Synopsis

# SYNOPSIS

The goal of the project was to establish a website where the customer can buy books of his choice with just a few mouse-clicks. The customer can browse the site to find a book he needs. If he finds it to be at a reasonable price, he has to register himself and buy it.

## The Steps:

- The user enters the website by specifying the URL in the web browser.
- He can browse the site to see what are the various books on sale without registering himself.
- When he finds a book he wants to buy, he has to register himself.
- Once he is registered, he gets a "Cart" for himself where he keeps the books of his choice.
- After making his selection, he views his cart.
- The bill is displayed.
- At this point, if he wants to remove any book from the cart for whatsoever reason, he may do so.
- Once he confirms his order, all the details are stored in order to deliver the books to him.
- At all points, the user can browse for more books.
- When he logs out, the allotted cart is withdrawn.
- The user is given a unique identity by which he is recognized every time he logs on to the website.
- All payments are done with a credit card.

- A Search Engine called "BOOGLIE" is included. Here the user specifies either the name of the book or the author of the book that is searching for.
- If the book is on the stands, then the details are displayed.
- He can search for a book by specifying the category too.
- The customer can return to the home page at any point.
- He may log out too at any point.

## INTRODUCTION

E Commerce is the emerging technology employed in the web. Our B2C site www.bookworld.com enables users to buy books on the net. User without credit cards can also browse our site for reviews on the latest books.

Platform-independence is a must for all web-based applications. Hence, "bookworld" has been implemented in Java and therefore enjoys all the advantages of "WRITE ONCE RUN ANYWHERE".

Servlets have been used rather than CGI due its advantages like

- Robustness
- Multithreading rather than multi processing
- Persistence so that they need to be loaded only once
- Platform-independence

# Conceptual Perspective

## JAVA - THE WEBMASTER

Just as popular World Wide Web browser software like Netscape has transformed the Web from a scientist's research tool into a consumer medium over the last two years...the Java programming language will transport the Web to the next level.

Hot Java could help both sound and animation performance levels reach new levels that would allow developers to create impressive Web sites.

Hot Java, the groundbreaking Web browser from Sun Microsystems, is earning unqualified kudos for its potential to elevate online computing to unprecedented heights. Co-written by a Sun insider, this expert guide to Hot Java and the Java programming language provides the first authoritative examination of this amazing new technology.

Few programming languages have generated as much excitement as Java. Used in conjunction with Hot Java, it enables Web site visitors for the first time to run small applications while still connected to the site. The possibilities are limitless.

Java is a blend of the best elements of its rich heritage combined with the innovative concepts required by its unique environment. Java derives its syntax from C and most of its object-oriented features from C++. Although Java has become inseparably linked with the online environment, it is a programming language.

The creation of Java was driven by 2 elements:

- To adapt to changing environment and uses

- To implement refinements and improvements in the art of programming

The Internet helped catapult Java to the forefront of programming, and Java in turn has had a profound effect on the Internet. The reason for this is quite simple: Java expands the universe of objects that can move about freely in the cyberspace.

In a network, 2 very broad categories of objects are transmitted between the server and the user's personal computer:

1. Passive information
2. Dynamic/Active Programs

Passive information does not change during the course of time period.
E.g.: email.
Active information is a dynamic, self-executing program that changes in accordance with the user's input.

The key that allows Java to solve both the security and the portability problems just described is that the output of a Java compiler is not executable code. Rather, it's a *Bytecode*. Bytecode is a highly optimized set of instruction designed to be executed by a Java run-time system, which is known as *Java Virtual Machine* (*JVM*). Java Virtual Machine is an interpreter for bytecode that makes it much easier to run a program in different environments.

Although the fundamental forces that necessitated the invention of Java are portability and security, other factors that are the highlights of the language are

- Simplicity
- Object-oriented approach

- Robustness
- Multithreading
- Neutral Architecture
- Interpretability
- High performance
- Dynamic programming

Just as popular World Wide Web browser software like Netscape has transformed the Web from a scientist's research tool into a consumer medium over the last two years...the Java programming language will transport the Web to the next level."
HotJava could help both sound and animation performance levels reach new levels that would allow developers to create impressive Web sites.

HotJava--the groundbreaking Web browser from Sun Microsystems--is earning unqualified kudos for its potential to elevate online computing to unprecedented heights. Co-written by a Sun insider, this expert guide to HotJava and the Java programming language provides the first authoritative examination of this amazing new technology.

Few programming languages have generated as much excitement as Java. Used in conjunction with HotJava, it enables Web site visitors for the first time to run small applications while still connected to the site. The possibilities are limitless.

## THE NEED FOR SERVLETS

Servlet technology provides web developers with a simple, consistent mechanism for extending the functionality of a web server and for accessing existing business systems. A servlet can almost be thought of as an applet that runs on the server side -- without a face. Java servlets have made many web applications possible.

Servlets are the Java platform technology of choice for extending and enhancing web servers. Servlets provide a component-based, platform-independent method for building web-based applications, without the performance limitations of CGI programs. And unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server and platform-independent. This leaves you free to select a "best of breed" strategy for your servers, platforms, and tools.

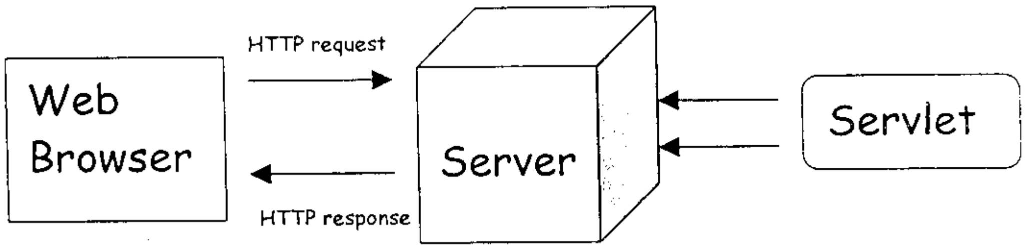
Servlets have access to the entire family of Java APIs, including the JDBC API to access enterprise databases. Servlets can also access a library of HTTP-specific calls and receive all the benefits of the mature Java language, including portability, performance, reusability, and crash protection.

Servlets are loaded and executed by a web browser in the same manner that applets are loaded and executed.

The following list describes the basic flow while using servlets

- The client (most likely the web browser) makes a request via HTTP.

- The web browser receives the request and forwards it to the servlet. If the servlet has not been loaded yet, the web browser loads it into the Java Virtual Machine and executes it.
- The servlet receives the HTTP request and performs server process.
- The servlet then returns a response back to the web browser.
- The web browser performs the response to the client.



In the early days, a server could dynamically construct a page by creating a separator process to handle each client request. The process would open connections to one or more databases in order to obtain the necessary information. It communicates with the web browser via the interface known as the *Common Gateway Interface* (CGI). CGI allows the separate processes to read from the HTTP request and write data to the HTTP response. A verity of different language was used to build the CGI programs, including C, C++ and PERL.

However, CGI suffered serious performance problem. Creating a separate process for each client request was expensive, in terms of processor and memory resources. Hence, other techniques were introduced.

Servlets have the following features:

- Servlets are persistent. Servlets are loaded only once by the web browser and maintain services between requests.

- Servlets are fast. Since Servlets only need to be loaded once, they offer much better performance.
- Servlets are platform independent since they are written in JAVA
- Servlets are secure.

## Overview of Java Servlets

- A servlet is a Java program that is run by a Web server.
- Servlets follow a standard servlet API defining the interface between the server and the servlet, and are designed to work within a request/response model. The servlet API is part of the standard extensions of JDK.
- Servlets can provide all the services of standard CGI, but are platform-independent, i.e. they can be used with any server implementing the API.
- More generally, servlets play the role of providing middle-tier services between the client and the back-end applications.

## Temporary and Permanent Servlets

- A temporary servlet is started when a request arrives and shut down after the response is generated.
- A permanent servlet is loaded when the server is started and lives until the server is shut down.

## The Servlet Life Cycle

- On the web server host machine, servlets run under the same process as the web server.
- The web server is responsible for creating an instance of the servlet and invoking standard methods from the interface (in a similar way to

a browser invoking methods of an applet). There are three main methods:

- init ( )
- service ( )
- destroy ( )

and two helper methods:

- getServletInfo ( )
- getServletConfig ( )

## <u>The destroy method</u>

- Called when the servlet in unloaded to clean up any open resources.
- Although the server normally waits until all service calls are terminated to invoke destroy, it may not be possible, and your destroy method should make sure that resources are not being used.

# MS ACCESS AT A GLANCE

In other database management systems, the term database is used to refer to tables that hold data. Access uses the term more broadly. An Access database consists of tables that hold data and all the related objects, such as queries, forms and reports that used to manage the data.

When you open a database, Access displays the database window, sometimes called the *database container*, because it contains all the objects that contain the database.

We can create tables in the design view or using the typical wizard. We also have the provision to create forms. Forms let us control the format in which the data gets displayed on the screen.

## Salient Features:

- Macros that let you automate and speed up your work.

  Macro is a list of actions. Access performs all the actions when the macro is run.

- Other utilities such as
  - Creating windows shortcuts.
  - Using hyperlink data type.
  - Creating web pages.
  - Creating indexes based on single or multiple fields.
  - Working on both embedded and linked OLE objects in Access, tables and queries.
  - Working with bound and unbound OLE objects in forms and reports

- Attaching a table from another database application so that Access and the other application can use it simultaneously.
- Customizing the Access working environment using the option dialogue box.

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

## JAVA WEB SERVER

Java Web Server product offers a cost-effective, easy-to-use, highly flexible, secure, and platform-independent solution that is designed to increase the speed of Internet and Intranet communications by simplifying the deployment and management of your web sites. It unites the versatility and security of the Java programming language with the ubiquity of the Internet.

Based on the Java programming language, the Java Web Server 2.0 product uses Java technology servlets -- server side programs -- that enable easy extensions to the power of the web server. Servlets take the place of cumbersome CGI scripts. They even run old CGI scripts. Servlets also give developers an interface that can be used on any platform (running a Java virtual machine) without additional porting.

The Java Web Server 2.0 product installs quickly. GUI-based tools provide a single point-of-control and make it painless to tune and manage a web site. Once configured, the Java Web Server product is easy to administer and there is no need to bring down the system to reconfigure the server.

The Java Web Server product combines the strength of SSL-compliant encryption and authentication technology with the advantages of the Java platform security model. Servlets, like applets, can be isolated into predefined, secure areas - sandboxes - that significantly reduce the enterprise's exposure to security risk. With the Java Web Server 2.0

product, local servlets are trusted and remote servlets can be assured of coming from true and trusted sources through digital signatures. Access Control lists (ACLs) are used to control which authenticated user gets access to specific web site resources

## About HTML

HTML or Hyper Text Markup Language is designed to specify the logical organization of a document, with important hypertext extensions. It is *not* designed to be the language of a WYSIWYG word processor such as Word or WordPerfect. This choice was made because many different "browsers", of very different abilities, may view the same HTML document. Thus, for example, HTML allows you to mark selections of text as titles or paragraphs, and then leaves the interpretation of these marked *elements* up to the browser. For example one browser may indent the beginning of a paragraph, while another may only leave a blank line.

HTML instructions divide the text of a document into blocks called *elements*. These can be divided into two broad categories -- those that define how the BODY of the document is to be displayed by the browser, and those that define information `about' the document, such as the title or relationships to other documents.

The detailed rules for HTML (the names of the tags/elements, how they can be used) are defined using another language known as the Standard Generalized Markup Language, or SGML. SGML is wickedly difficult, and was designed for massive document collections, such as repair manuals for F-16 fighters, or maintenance plans for nuclear submarines. Fortunately, HTML is much simpler!

However, SGML has useful features that HTML lacks. For this reason, markup language and software experts have developed a new language, called

XML (the *eXtensible markup language*), which has most of the most useful features of HTML and SGML.

## Elements in HTML Documents

The HTML instructions, along with the text to which the instructions apply, are called HTML *elements*. The HTML instructions are themselves called *tags*, and look like <element_name> -- that is, they are simply the element name surrounded by left and right angle brackets.

Most elements mark blocks of the document for particular purpose or formatting: the above <element_name> tag marks the beginning of such as section. The end of this section is then marked by the *ending* tag </element_name> -- note the leading slash character "/" that appears in front of the element name in an end tag. This leading slash character always indicates end, or stop tags.

### Empty Elements

Some elements are *empty* -- that is, they do not affect a block of the document in some way. These elements do not require an ending *tag*. An example is the <HR> element, which draws a horizontal line across the page. This element would simply be entered as <HR>

### Upper and Lower Case

Element names are case *insensitive*. Thus, the horizontal rule element can be written as any of <hr>, <Hr> or <HR>.

### Elements can have Attributes

Many elements can have arguments that pass parameters to the interpreter handling this element. These arguments are called *attributes* of the element. For example, consider the element A, which marks a region of text as the beginning (or end) of a hypertext link. This element can have several attributes. One of them, HREF, specifies the hypertext document to which the marked piece of text is linked. To specify this in the tag for A you write:

<A HREF="http://www.somewhere.ca/file.html"> marked text </a>.

where the attribute HREF is assigned the indicated value. Note that the "A" element is not empty, and that it is closed by the tag </a>. Note also that end tags *never* take attributes -- the attributes to an element are always placed in the start tag.

## HTML Document Structure

HTML documents are structured into two parts, the HEAD, and the BODY. Both of these are contained within the HTML element -- this element simply denotes this as an HTML document.

The head contains information about the document that is not generally displayed with the document, such as its TITLE. The BODY contains the body of the text, and is where you place the document material to be displayed. Elements allowed inside the HEAD, such as TITLE, are not allowed inside the BODY, and vice versa. When your HTML browser (Netscape Navigator, Internet Explorer, Opera, lynx etc) retrieves a file, it must know what type of data it has received in order to know what to do with it. Hypertext (that is, HTTP) servers explicitly tell the browser the type of the data being sent. In other cases, such as when the browser is using FTP to access a remote file, or when the browser is reading a file from your local disk (such as when you are editing pages prior to publishing them to a Web server), the browsers "guesses" the data type from the filename extension -- that is the part after the dot in the filename. For example, HTML files are identified by names such as name.html, where the .html extension indicates an HTML document.

# Implementation Details

# B2C and e-Commerce

Our site provides shared membership information, multiple shopping carts, inventory database, dynamically generated web pages, feature comparison, choice of carriers, freight, sales tax and credit card processing.

The primary difference between B2B and B2C is that B2B has limited Intranet access limite those companies and employees with accounts pre-assigned by the site administrator. Meth of payment would typically be electronic funds transfer for B2B and credit cards for B2C. Mo of transportation for B2B would be expanded to include commercial freight, rail, sea, etc. as compared to the traditional methods of shipment for B2C. B2B can typically include a comm of inter-related users whereas B2C is typically a single consumer and a group of storefronts ( selling goods and services to individual consumers. However, the technology is very similar both B2B and B2C.

## ❖ Forms– Standardized with Error Checking

With today's web site development tools, it is very easy to insert a standard form on a web page. However, approximately 90% of the design work for creating custom forms is related to error checking of the data entered to insure that all required fields are complete and in a format that can be interpreted by the application software. If this error checking is not included, the recipient will receive many unusable applications and spend many hours correcting the form and sorting through meaningless data that has been be submitted. Failure to resolve the problem leaves the customer with the impression that the merchant is not responsive and their business is not important.

All forms are fill-in-the-blank with color-coded fields to distinguish between required and optional entries. When "Submit" is initiated, error checking is performed to verify that all required fields have been entered and that data is in a usable format. If any errors are detected, the respective fields will be color-coded yellow. The following error checks are performed:

## ❖ Security

Several forms of security would be provided on operational sites that are not in place on the demonstration site:

- All communications by secure server with up to 128-bit encryption depending on the user's browser.
- For B2B Intranet applications, electronic "firewalls' are used to restrict access to authorized users.
- All users have their own unique user name and password.
- In the event fraudulent access is suspected, trained personnel disable access until verbal verification is obtained.
- Logons time out after a predetermined amount of time without activity.

## ❖ Visibility

The Internet is a vast space and you can have the best web site in the universe. However, if customers don't know your address, you will be very disappointed in the results.

Just as "location, location, and location" are the three most important words in real estate, "visibility, visibility and visibility" are the equivalent for any web site and especially for an e-commerce site where the merchant is not restricted by geographic boundaries.

- All fields are pre-defined as alphanumeric, alpha, or numeric. Error checking then limits valid entries to the pre-defined format. For example, if you want to allow entering a phone number as 800-Flowers, the phone number field was be defined as alphanumeric. The phone number is defined as numeric in our demonstration.
- Valid 2-character state code
- Zip code of 5 to 9 numeric digits
- Telephone number with 10 numeric digits
- E-mail address includes a "@" and a "dot" with one or more alphanumeric characters to the right of the dot. This could be further restricted to require 3 alpha characters or specified extensions of "com", "net", "org", "edu" but then considers international extensions as invalid.

For user convenience, a carriage return after an entry is treated as a "Submit". In general application to domestic operations can be more restrictive than international sites.

## ❖ New User and Member Logon

New users are transferred to "Member Services" to complete registration. Existing members simply enter their e-mail address (or Customer ID) and password and commence shopping.

## ❖ Member Services

### ➢ Personal Data

Minimum registration data is entered on this form, which consists of name, address, telephone, and password. For existing members this data will be displayed as soon as they logon and can be edited. Addresses are "Bill To" addresses and must agree with the address on the credit card(s).

### ➢ Payment Data

All transactions are paid via credit cards for B2C and accounts / electronic funds transfers for B2B. Members can enter data for multiple credit cards and select one card to be their "default" card.

Users have the option of deleting or retaining credit card data when they log off.

## ❖ Shopping Cart Selection

Each user can have a shopping cart. This is implemented with a customer database and does not require "cookies" to retain the status of the shopping card from previous visits. This allows routine orders to be left in a shopping cart for quick recall and placement of future repetitive orders. After logging in, any existing shopping carts are initialized to their status as of the last visit.

Users can view their shopping carts at any time to check the total of their purchases, delete items, or edit their address book and/or credit card.

### ❖ Add to cart

An "add to cart" button is provided beside each item. When initiated the item is added to the user's shopping cart. This button changes to "Notify" for items that are out of stock. Members can request to be notified by e-mail when the item is available for shipment.

## ❖ Administration

Individual merchants control inventory and price updates via Intranet. Updates can be manually entered, edited and deleted or automatically downloaded from distributors and other vendors. Product class and / or price, as either a percentage or fixed dollar value, determine markups for customer orders.

All sales data can be integrated with the accounting system of the individual merchants to eliminate the need for manual entry into their accounting system software.

## ❖ Security

Several forms of security would be provided on operational sites that are not in place on the demonstration site:

- All users have their own unique user name and password.
- In the event fraudulent access is suspected, trained personnel disable access until verbal verification is obtained.
- Logons time out after a predetermined amount of time without activity.

## ❖ Visibility

The Internet is a vast space and you can have the best web site in the universe. However, if customers don't know your address, you will be very disappointed in the results.
Just as "location, location, and location" are the three most important words in real estate, "visibility, visibility and visibility" are the equivalent for any web

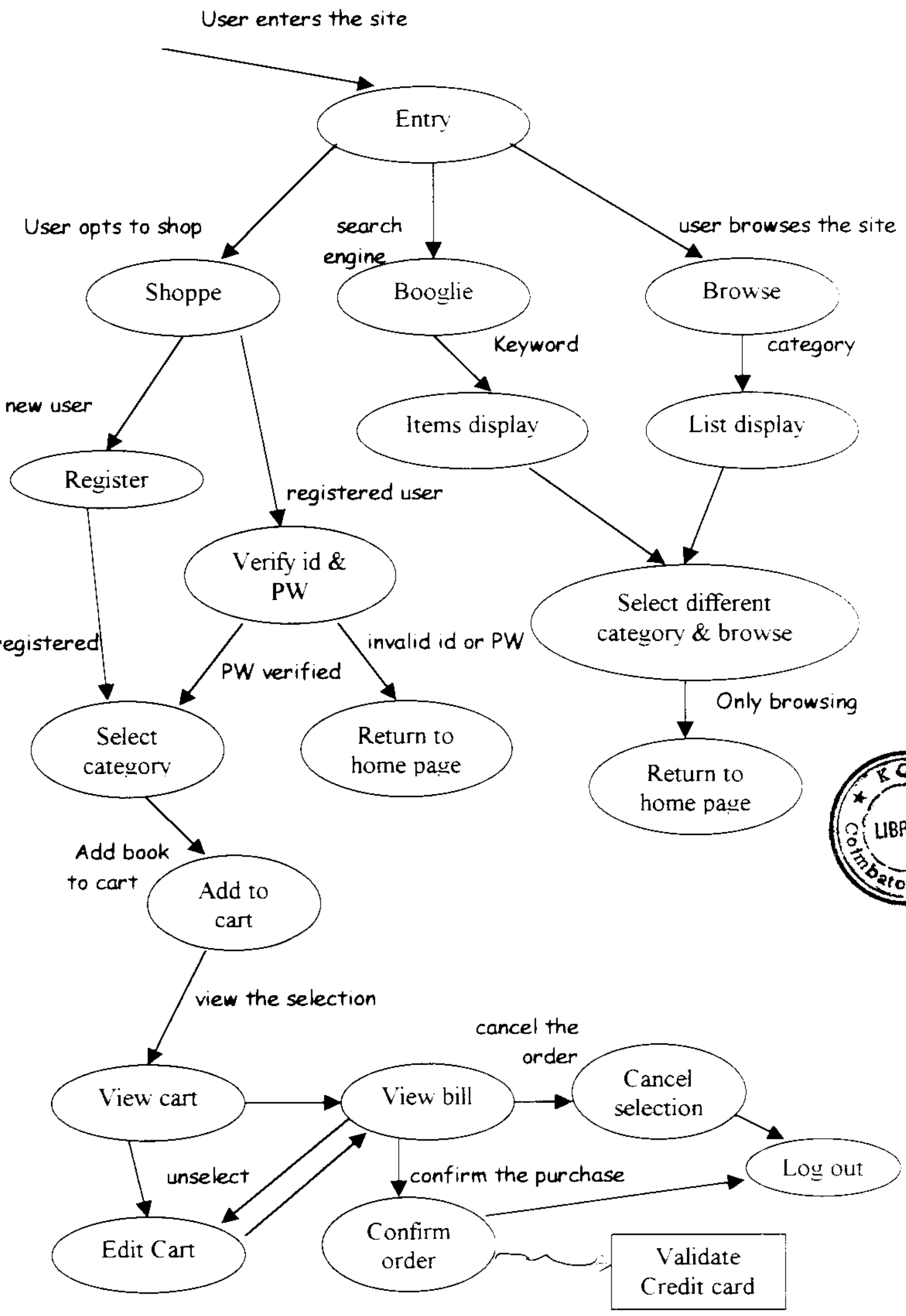
site and especially for a e-commerce site where the merchant is not restricted by geographic boundaries.

## ➢ Check Out

When the user is ready, they can proceed to checkout. Each shopping cart is processed as a separate transaction and is typically associated with different methods of payment and / or shipping destinations.

On-line payment verification is included to minimize the administration and facilitate customer purchases.

# DATA FLOW DIAGRAM



# Database Design

| | |
|---|---|
| userid | Id of the user |
| password | password |
| name | Name of the user |
| email | Email address |
| address | Residential address |
| phone | Phone number |
| creditno | Credit card number |
| compocc | Company of credit |

## Book information

| | |
|---|---|
| Book_Id | Id of the book |
| Book_Name | Name of the book |
| Book_Author | Author of the book |
| Price | Price of the book |
| Copies | Copies available |
| Category | Category of the book |
| Keyword | keyword |
| Preview | Preview |
| | |

**Cart information**

| | |
|---|---|
| Id | Id of the user |
| Bcode | Code of the book |
| Copies | Copies available |

# Appendix

# Sample Coding

## Administrator Updation

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

 public class admin extends HttpServlet
 {    PreparedStatement st;
      ResultSet  rs;
      Connection con;
      String temp=null;
   public void init(ServletConfig cfg) throws ServletException
        {  super.init(cfg);
           try
           {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con= DriverManager.getConnection("Jdbc:Odbc:project","","");
           }
            catch(Exception e)
             {
              e.printStackTrace();
             }
        }

   public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<html><head>");
        out.println("<meta http-equiv= "Content-Type " content= "text/html;
charset=iso-8859-1\">");
        out.println("<meta name=\"GENERATOR \" content=\"Microsoft
FrontPage Express 2.0\">");
        out.println("<title>ADMINISTRATOR</title></head>");
        out.println("<body bgcolor=\"#FFFFFF \">");
        out.println("<table border= "1 " width= "100% \">");
        out.println("<tr>");
        out.println("<td><B>BOOKID</B></td>");
        out.println("<td><b>BOOKNAME</B></td>");
out.println("<td><B>AUTHOR</B></td>");
out.println("<td><B>COST</B></td>");
out.println("<td><B>NO.OF COPIES</td>");
out.println("</tr>");
String tempad=req.getParameter("admin");
String tempw=req.getParameter("pw");
try
{st = con.prepareStatement("select * from admin where AName=?");
st.setString(1,tempad);
 rs=st.executeQuery();
 while(rs.next())
 temp=rs.getString(2);
 if(temp.equals(tempw))
   out.println(temp+"  U r permitted!!!!!");
 }
   catch(Exception e)
      {e.printStackTrace();
      }
 }
}
```

## Administrator

```
import  javax.servlet.*;
import  javax.servlet.http.*;
import java.io.*;

public class administer extends HttpServlet {

     public void  doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
    {
       res.setContentType("text/html");
       String user =req.getParameter("user");
       PrintWriter out  = res.getWriter();
       out.println("<html>");
       out.println("<head>");
       out.println("<title> Administer Login </title>");
       out.println("</head>");
```

**Administrator Updation**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

 public class admin extends HttpServlet
 {    PreparedStatement st;
     ResultSet  rs;
     Connection con;
     String temp=null;
   public void init(ServletConfig cfg) throws ServletException
       {   super.init(cfg);
          try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          con= DriverManager.getConnection("Jdbc:Odbc:project","","");
          }
           catch(Exception e)
            {
            e.printStackTrace();
            }
       }

   public void doGet(HttpServletRequest req,HttpServletResponse res)
 throws ServletException,IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        out.println("<html><head>");
        out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=iso-8859-1\">");
        out.println("<meta name=\"GENERATOR\" content=\"Microsoft
 FrontPage Express 2.0\">");
        out.println("<title>ADMINISTRATOR</title></head>");
        out.println("<body bgcolor=\"#FFFFFF\">");
        out.println("<table border=\"1\" width=\"100%\">");
        out.println("<tr>");
        out.println("<td><B>BOOKID</B></td>");
        out.println("<td><b>BOOKNAME</B></td>");
        out.println("<frameset cols=\"20%,80%\">");
        out.println("<frame src=\"\\servlet\option?user="+user+"\" 
name=\"one\">");
        out.println("<frame src=\"c:\\97cse38\\frame3.html\" 
name=\"two\">");
        out.println("</frameset>");
        out.println("</html>");
      }
        }
}
```

**List Display**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

 public class admin extends HttpServlet
 {    PreparedStatement st;
     ResultSet  rs;
     Connection con;
     String temp=null;
  public void init(ServletConfig cfg) throws ServletException
       {  super.init(cfg);
          try
          {
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con= DriverManager.getConnection("Jdbc:Odbc:project","","");
          }
           catch(Exception e)
           {
            e.printStackTrace();
           }
       }
  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
       res.setContentType("text/html");
       PrintWriter out = res.getWriter();
       out.println("<html><head>");
       out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=iso-8859-1\">");
       out.println("<meta name=\"GENERATOR\" content=\"Microsoft
FrontPage Express 2.0\">");
       out.println("<title>ADMINISTRATOR</title></head>");
       out.println("<body bgcolor=\"#FFFFFF\">");
       out.println("<table border=\"1\" width=\"100%\">");
       out.println("<tr>");
       out.println("<td><B>BOOKID</B></td>");
       out.println("<td><b>BOOKNAME</B></td>");
       out.println("<td><B>AUTHOR</B></td>");
       out.println("<td><B>COST</B></td>");
       out.println("<td><B>NO.OF COPIES</td>");
       out.println("</tr>");
       String tempad=req.getParameter("admin");
       String tempw=req.getParameter("pw");
       try
       {st = con.prepareStatement("select * from admin where AName=?");
       st.setString(1,tempad);
        rs=st.executeQuery();
        while(rs.next())
        temp=rs.getString(2);
        if(temp.equals(tempw))
          out.println(temp+"  U r permitted!!!!!");
        }
          catch(Exception e)
            {e.printStackTrace();
            }
        }
       }
```

**BILL**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

 public class bill extends HttpServlet
 {    PreparedStatement st;
      ResultSet  rs;
      Connection con;
      String temp=null;
   public void init(ServletConfig cfg) throws ServletException
        {  super.init(cfg);
           try
           {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con= DriverManager.getConnection("Jdbc:Odbc:project","","");
           }
            catch(Exception e)
            {
             e.printStackTrace();
            }
        }

   public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        String id =req.getParameter("id");
        String tcount =req.getParameter("count");
        String user =req.getParameter("user");
        int count = Integer.parseInt(tcount);
        try
        {
             st=con.prepareStatement("insert into Cart(Id,Bcode,copies) values
(?,?,?)");
             st.setString(1,user);
             st.setString(2,id);
             st.setInt(3,count);
             st.executeUpdate();
             out.println("updated successfully");
//             st =con.prepareStatement("update COPIES BOOK_INFO
        }
        catch(Exception e)
        {
        }
        out.println("<a href =\"\\servlet\\basket?user="+user+"\">view
cart</a>");
        out.println(" ");
        }
        }
```

**Cancellation of order**

```
import java.io.*;
import javax.servlet.*;
import java.util.*;
import javax.servlet.http.*;
import java.sql.*;


public class cancel extends HttpServlet {
        PreparedStatement st;
        ResultSet  rs;
        PreparedStatement st1;
        ResultSet  rs1;

        Connection con;
        int c;
  public void init(ServletConfig cfg) throws ServletException
          {
          super.init(cfg);
          try
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         con= DriverManager.getConnection("Jdbc:Odbc:project","","");
           }
         catch(Exception e)
          {
          e.printStackTrace();

          }
```

```
    }


  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
    {
     res.setContentType("text/html");
    PrintWriter out = res.getWriter();
     String user=null;
      user = req.getParameter("user");

out.println("<html>");
out.println("<head>");
out.println("<title>");
out.println("</title>");
out.println("</head>");
out.println("<body>");
out.println("<font size=+2>");
out.println("<body  bgcolor=white>");




        try {
          st = con.prepareStatement("delete * from Cart where Id=?");
          st.setString(1,user);
          st.executeUpdate();
          }

          catch(Exception e)
             {
           e.printStackTrace();
          }
        out.println("your order is cancelled");
        }}
```

**Addition to cart**

```
import java.io.*;
import javax.servlet.*;
import java.util.*;
import javax.servlet.http.*;
import java.sql.*;


public class cart extends HttpServlet {
          PreparedStatement st;
          ResultSet  rs;
          PreparedStatement st1;
          Connection con;
  public void init(ServletConfig cfg) throws ServletException
            {
            super.init(cfg);
            try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con= DriverManager.getConnection("Jdbc:Odbc:project","","");
            }
           catch(Exception e)
            {
            e.printStackTrace();

            }
        }


  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
     {
      res.setContentType("text/html");
     PrintWriter out = res.getWriter();
      String user=null;
      String tcount=null;
```

```
        String bcode=null;
        int count;
        int j=0;
         user = req.getParameter("user");
        tcount = req.getParameter("count");
          String type = req.getParameter("type");

        count=Integer.parseInt(tcount);
        String check[]= new String[count+1];
        String tcheck[] = new String[count+1];
        count = Integer.parseInt(tcount);
        int a[] =new int[count+1];
              for(int i=1;i<=count;i++)
        {
              a[i]=0;

              tcheck[i]="on";
        }
        for(int i=1;i<count+1;i++)
        {

              check[i]=req.getParameter(""+i+"");
              System.out.println(check[i]+"   "+tcheck[i]);
              if(tcheck[i].equals(check[i]))
                          a[i]=1;
              else
                          a[i]=0;
        }




out.println("<html>");
out.println("<head>");
out.println("<title>");
out.println("</title>");
out.println("</head>");
out.println("<body>");
out.println("<form >");
out.println("<font size=+2>");
out.println("<body  bgcolor=white>");



        try {
          st = con.prepareStatement("select * from BOOK_INFO where
category =?");
          st.setString(1,type);
          rs=st.executeQuery();
          System.out.println("error1");
          out.println("<body>");
          out.println("<table border = \"1 \">");

out.println("<tr><td>SNO</td><td>BookId</td><td>Name</td><td>Autho
r</td><td>copies</td></tr>");
          while(rs.next())
          {
          System.out.println("error2");
          j++;
          if(a[j]==1)
          {
            out.println("<tr>");
           out.println("<td>"+j+"</td>");
           bcode=rs.getString(1);
           out.println("<td>"+bcode+"</td>");
           String title1=rs.getString(2);
           out.println("<td><a href
=\"\\servlet\\view1?user="+user+"&id="+bcode+"\">"+title1+"</a></td>");
           out.println("<td>"+rs.getString(3)+"</td>");
            out.println("</tr>");
                  }
       }
           out.println("</table>");
           }
          catch(Exception e)
             {
           e.printStackTrace();
          }
out.println("</body>");
out.println("</html>");
              }          }




import java.util.Enumeration;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class ccc extends HttpServlet
{
   HttpServletRequest req;
   HttpServletResponse res;
   boolean flag = false;
   String user = null;
   String tcount = null;
   String pass = null;
   String tqty = null;
   String tcdno = null;
   String bcode = null;

   int qty=0;
   int cdno=0;
   float amt;
   float total;
   float price;
   int count = 0;
   String type=null;
   int j;


   Connection con;
   PreparedStatement st;
   ResultSet rs;
   PreparedStatement st1;
   ResultSet rs1;
   PreparedStatement st2;
```

```
ResultSet rs2;

PrintWriter out;

public void init(ServletConfig c) throws ServletException
{
   super.init(c);
   try
   {
   Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
   con=DriverManager.getConnection("jdbc:odbc:project");

   }
   catch(Exception e)
   {
   }
}

public void doGet(HttpServletRequest requ,HttpServletResponse resp)
throws ServletException,IOException
{
   count = 0;
   total=0;
   flag = false;
   req = requ;
   res = resp;
   amt=0;
   int c=0;
   System.out.println("hai");
   user = req.getParameter("userId");
   tcount = req.getParameter("count");
   count=Integer.parseInt(tcount);
   type = req.getParameter("type");

   String check[]= new String[count+1];
   String tcheck[] = new String[count+1];
   count = Integer.parseInt(tcount);
   int a[] =new int[count+1];

   for(int i=1;i<=count;i++)
```

```
    {
        a[i]=0;

        tcheck[i]="on";
    }
    for(int i=1;i<count+1;i++)
    {

         check[i]=requ.getParameter(""+i+"");
         System.out.println(check[i]);
         if(tcheck[i].equals(check[i]))
                   a[i]=1;
         else
                   a[i]=0;
    }


       resp.setContentType("text/html");
       out = resp.getWriter();
       out.println("<HTML><TITLE>SHOPPING CART</TITLE>");
       out.println("<body bgcolor = \"000000\" text = \"00FF00\" >");
       out.println("<head><h1><font color = \"FF0000\">CUSTOMER
CART DETAILS</font></h1></head>");
     try
     {

       out.println("          <body bgcolor=\"#000000\"><center>");





         st = con.prepareStatement("select * from BOOK_INFO where
category =?");
         st.setString(1,type);
         rs=st.executeQuery();
         out.println("<body>");
         out.println("<table border =\"1\">");
out.println("<tr><td>SNO</td><td>BookId</td><td>Name</td><td>Autho
r</td><td>copies</td></tr>");
           out.println("<form action =\"\servlet\cart\" method =\"get\">");
         while(rs.next())
         {
         j++;
         if(a[j]==1)
         {
           out.println("<tr>");
          out.println("<td>"+j+"</td>");
          bcode=rs.getString(1);
          out.println("<td>"+bcode+"</td>");
          out.println("<td>"+rs.getString(2)+"</td>");
          out.println("<td>"+rs.getString(3)+"</td>");
          out.println("<td><input type =\"text\" name =\""+j+"\" value
=1></td>");

           out.println("</tr>");
           st1=con.prepareStatement("insert into Cart (Id,bcode,copies)
values  (?,?,?)");
           st1.setString(1,user);
           st1.setString(2,bcode);
           st1.setInt(3,1);
           st1.executeUpdate();
                 }
      }
          out.println("</table>");
          out.println("<input type =\"submit\"  value =\" add to cart\">");
          out.println("<input type =\"hidden\"  name = \"count\" value
=\""+j+"\">");
          out.println("<input type =\"hidden\"  name = \"type\" value
=\""+type+"\">");
          out.println("<input type =\"hidden\"  value =\""+user+"\">");
          out.println("</form>");
          }
         catch(Exception e)
           {
          e.printStackTrace();
         }
```

```
        out.println("</center></body></html>");




        }
}
```

**User Confirmation**

```
import java.io.*;
import javax.servlet.*;
import java.util.*;
import javax.servlet.http.*;
import java.sql.*;


public class confirm extends HttpServlet {
        PreparedStatement st;
        ResultSet  rs;
        PreparedStatement st1;
        ResultSet  rs1;

        Connection con;
        int c;
  public void init(ServletConfig cfg) throws ServletException
          {
          super.init(cfg);
          try
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         con= DriverManager.getConnection("Jdbc:Odbc:project","","");
          }
         catch(Exception e)
          {
            e.printStackTrace();

            }
        }


  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
     {
      res.setContentType("text/html");
     PrintWriter out = res.getWriter();
      String user=null;
       user = req.getParameter("user");

out.println("<html>");
out.println("<head>");
out.println("<title>");
out.println("</title>");
out.println("</head>");
out.println("<body>");
out.println("<font size=+2>");
out.println("<body  bgcolor=white>");




        try {
          st = con.prepareStatement("select * from Cart where Id=?");
          st.setString(1,user);
          rs =st.executeQuery();
          while(rs.next())
          {
            st1 =con.prepareStatement("insert into con(Id,Bcode,copies)
values(?,?,?)");
            st1.setString(1,rs.getString(1));
            st1.setString(2,rs.getString(2));
            st1.setInt(3,rs.getInt(3));
            st1.executeUpdate();
        }
        st = con.prepareStatement("delete * from Cart where Id=?");
        st.setString(1,user);
        st.executeUpdate();

        }

        catch(Exception e)
          {
         e.printStackTrace();
        }
      out.println("your order is confirmed");
    }}
```

**Registration**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class detail extends HttpServlet {
          Statement st;
          ResultSet  rs;
          Connection con;
  public void init(ServletConfig cfg) throws ServletException

            {
            super.init(cfg);
            try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con= DriverManager.getConnection("Jdbc:Odbc:project","","");
             }
           catch(Exception e)
            {
            e.printStackTrace();

            }
```

```
      }
  public void doPost(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
      res.setContentType("text/html");
      PrintWriter out = res.getWriter();
       String Id=req.getParameter("uniq");
       String PW=req.getParameter("pw");
       System.out.println(PW);
       String Name=req.getParameter("uname");
       String email=req.getParameter("mail");
       System.out.println(email);
       String Address=req.getParameter("add");
       String Ph=req.getParameter("ph");
       String CreditCNo=req.getParameter("CC");
       String ComOCC=req.getParameter("comp");
       try {
       st=con.createStatement();
          if(PW==null || Id==null)
          {  out.println("enter all the details");
                    }
          st.executeUpdate("insert into user values
('"+Id+"','"+PW+"','"+Name+"','"+email+"','"+Address+"','"+Ph+"','"+Credit
CNo+"','"+ComOCC+"')");
         out.println("<h2>YOU HAVE BEEN SUCCESSFULLY
REGISTERED</h2>");
         //out.println("<a
href=\"\\servlet\\administer?user="+Id+"\">SHOPPING</a>");
          }
         catch(Exception e)
           {e.printStackTrace();
           }
          }
           }
```

**Selection for cart**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class list extends HttpServlet {
          PreparedStatement st;
          ResultSet  rs;
          Connection con;
  public void init(ServletConfig cfg) throws ServletException

            {
            super.init(cfg);
            try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con= DriverManager.getConnection("Jdbc:Odbc:project","","");
             }
           catch(Exception e)
            {
            e.printStackTrace();

            }
        }
   public void doGet(HttpServletRequest req,HttpServletResponse res)
 throws ServletException,IOException {
        res.setContentType("text/html");
        int i=0;
        PrintWriter out = res.getWriter();
         String type = req.getParameter("type");
         String user = req.getParameter("user");

         System.out.println(type);
         try {
           st = con.prepareStatement("select * from BOOK_INFO where
 category =?");
           st.setString(1,type);
           rs=st.executeQuery();
           out.println("<body>");
           out.println("<form action=\"\\servlet\\cart\" method = \"get\">");
           out.println("<table border =\"1\">");
out.println("<tr><td>SNO</td><td>ADD</td><td>BookId</td><td>Name
</td><td>Author</td></tr>");
          while(rs.next())
          {
          i++;
            out.println("<tr>");
           out.println("<td>"+i+"</td>");

           out.println("<td><input type = \"checkbox\" name
=\""+i+"\"></td>");
           out.println("<td>"+rs.getString(1)+"</td>");

           out.println("<td>"+rs.getString(2)+"</td>");
           out.println("<td>"+rs.getString(3)+"</td>");


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

          }
           out.println("</table>");
                      out.println("<input type = \"submit\"  value = \"      VIEW
\">");
            out.println("<input type =\"hidden\"  name = \"count\" value
=\""+i+"\">");
            out.println("<input type =\"hidden\"  name = \"type\" value
=\""+type+"\">");

            out.println("<input type =\"hidden\"  name =\"user\" value
=\""+user+"\">");


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

            }
           catch(Exception e)
             {e.printStackTrace();
                         ;
             }
             out.println("</body></html>");
```

```
        }}
```

**Browsing**

```
import  javax.servlet.*;
import  javax.servlet.http.*;
import java.io.*;

public class administer extends HttpServlet {

     public void  doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
    {
         res.setContentType("text/html");
         PrintWriter out  = req.getWrtiter();
         out.println("<html>");
         out.println("<head>");
         out.println("<title> Administer Login </title>");
         out.println("</head>");
         out.println("<frameset cols=\"20%,80%\">");
         out.println("<frame src=\"frame1.html name=\"one"\>");
         out.println("<frame src=\"frame2.html name=\"two"\>");
         out.println("</frameset>");
         out.println("</html>");
        }
          }



import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class option extends HttpServlet {
          PreparedStatement st;
          ResultSet  rs;
  public void init(ServletConfig cfg) throws ServletException
            {
```

```
            super.init(cfg);
        }
  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
        res.setContentType("text/html");
        String user =req.getParameter("user");
        PrintWriter out = res.getWriter();
out.println("<html><head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR\" content=\"Microsoft FrontPage
Express 2.0\">");
out.println("<title>Untitled Normal Page</title></head>");
out.println("<body bgcolor=\"#FFFFFF\">");
out.println("<br>");

out.println("<a href =\"\\servlet\\list?type=novel&user="+user+"\" target
=\"two\">NOVEL</a>");
out.println("<br>");

out.println("<a href =\"\\servlet\\list?type=technical&user="+user+"\" target
=\"two\">TECHNICAL</a>");
out.println("<br>");

out.println("<a href =\"\\servlet\\list?type=general&user="+user+"\" target
=\"two\">GENERAL</a>");
out.println("<br>");

out.println("<a href =\"\\servlet\\list?type=sports&user="+user+"\" target
=\"two\">SPORT</a>");
out.println("<br>");
out.println("<a href =\"\\servlet\\list?type=psychology&user="+user+"\"
target =\"two\">PSYCHOLOGY</a>");
out.println("<br>");
//out.println("<a href
="\\servlet\\list?type=novel&user="+user+"\">NOVEL</a>");
out.println("</body></html>");
}
}
```

**Booglie**

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;

public class search extends HttpServlet {
          PreparedStatement st;
          ResultSet  rs;
          Connection con;
  public void init(ServletConfig cfg) throws ServletException

            {
            super.init(cfg);
            try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
           con= DriverManager.getConnection("Jdbc:Odbc:project","","");
             }
           catch(Exception e)
            {
            e.printStackTrace();

            }
        }
  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException {
       res.setContentType("text/html");
       PrintWriter out = res.getWriter();
out.println("<html><head>");
out.println("<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=iso-8859-1\">");
out.println("<meta name=\"GENERATOR \" content=\"Microsoft FrontPage
Express 2.0\">");
out.println("<title>Untitled Normal Page</title></head>");
out.println("<body bgcolor=\"#FFFFFF\">");
out.println("<table border=\"1\" width= "100%\">");
      out.println("<tr>");
      out.println("<td><B>BOOKID</B></td>");
      out.println("<td><b>BOOKNAME</B></td>");
      out.println("<td><B>AUTHOR</B></td>");
      out.println("<td><B>COST</B></td>");
      out.println("<td><B>NO.OF COPIES</td>");
      out.println("</tr>");

        String temp=req.getParameter("search");
         try {
          st = con.prepareStatement("select * from BOOK_INFO where
KEYWORD=?");
          st.setString(1,temp);
          rs=st.executeQuery();

          while (rs.next())
          {
       out.println("<tr>");
       out.println("<td>"+rs.getString(1)+"</td>");
       out.println("<td><a
href=\"c:\\97cse38\\shoppe.html\">"+rs.getString(2)+"</a></td>");
       out.println("<td>"+rs.getString(3)+"</td>");
       out.println("<td>"+rs.getString(4)+"</td>");
       out.println("<td>"+rs.getString(5)+"</td>");
       out.println("</tr>");
            }
 out.println("</table></body></html>");
           }
           catch(Exception e)
             {e.printStackTrace();
             }
            }
           }
```

## Cart View

```
import java.io.*;
import javax.servlet.*;
import java.util.*;
import javax.servlet.http.*;
import java.sql.*;


public class view extends HttpServlet {
        PreparedStatement st;
        ResultSet  rs;
        PreparedStatement st1;
        Connection con;
  public void init(ServletConfig cfg) throws ServletException
           {
          super.init(cfg);
          try
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         con= DriverManager.getConnection("Jdbc:Odbc:project","","");
           }
         catch(Exception e)
          {
          e.printStackTrace();

          }
      }


  public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
     {
      res.setContentType("text/html");
     PrintWriter out = res.getWriter();
      String user=null;
      String tcount=null;
      String bcode=null;
      int count;
      int j=0;
       user = req.getParameter("user");
         String id = req.getParameter("id");


out.println("<html>");
out.println("<head>");
out.println("<title>");
out.println("</title>");
out.println("</head>");
out.println("<body>");
out.println("<form >");
out.println("<font size=+2>");
out.println("<body  bgcolor=white>");




        try {
          st = con.prepareStatement("select * from BOOK_INFO where
BOOK_ID =?");
          st.setString(1,id);
          rs=st.executeQuery();
          System.out.println("error1");
          out.println("<body>");
          out.println("<table border = "1 ">");
          while(rs.next())
          {
          System.out.println("error2");
            out.println("<tr>");
           out.println("<td>TITLE  :</td><td> "+rs.getString(2)+"</td>");
            out.println("</tr>");
             out.println("<tr>");

           bcode=rs.getString(1);
           out.println("<td>CODE   :</td><td> "+bcode+"</td>");
            out.println("</tr>");
           out.println("<tr>");

          out.println("<td>PRICE  :</td><td> "+rs.getString(3)+"</td>");
           out.println("</tr>");
            out.println("<tr>");

          out.println("<td>REVIEW :</td><td> "+rs.getString(8)+"</td>");

          System.out.println(user);
           out.println("</tr>");
      }
          out.println("</table>");
            out.println("<form action = \"/servlet/bill\" method = \"get\">");

          out.println("<input type = \"text\"  name = \"count\"   value =1 >");

          out.println("<input type = \"submit\"  value = \" add to cart\">");
          out.println("<input type = \"hidden\"  name = \"id\" value
=\""+bcode+"\">");
          out.println("<input type = \"hidden\"  name = \"user\" value
=\""+user+"\">");
          out.println("</form>");
          }
         catch(Exception e)
           {
          e.printStackTrace();
         }
out.println("</body>");
out.println("</html>");
            }           }
```

## HOME PAGE SOURCE CODE

```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>ONLINE SHOPPE</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-
Type>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY aLink=#ffffff bgProperties=fixed leftMargin=25 vLink=#000000
bgsound=""><STRONG>
<P>&nbsp;</P>
<FORM action=http://localhost:8080/servlet/search method=get
name=searchs>
<H2>WELCOME TO BOOKWORLD.COM
</H2></STRONG><I><FONT size=3>We are happy and
thankful for you for visiting our sight.
<U>BOOKWORLD</U>&nbsp;&nbsp;has&nbsp;great collection of books
which you can
purchase online .....</FONT></I> <BR><BR><FONT size=" ">LIST OF
TOPICS
</FONT><BR><BR>
<P align=center><U>NOVELS </U>| <U>ENTERTAINMENT </U>
<U><A
href="file:///C:/WINDOWS/Profiles/rajbabu/Desktop/proj/technical.html">
TECHNICAL</A>
</U><BR><U>SPORTS </U>| <U>POLITICS </U>| <U>SCIENCE
</U><BR><U>LITERATURE </U>
<U>ASTRONOMY </U>| <U>FICTION </U><BR><BR></P>We have
the greatest search engine
implemented here... The booglie search. <BR><BR>
<TABLE border=1>
  <TBODY>
  <TR>
    <TD>Search &nbsp;&nbsp;&nbsp; <INPUT name=search></TD>
    <TD><INPUT type=submit value=booglie
search></TD></TR></TBODY></TABLE></FORM>&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;
```

```
<P>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<B><FONT color=#0000ff face=Calligrapher size=2></FONT><FONT
color=#0000ff
face=Calligrapher size=4>JUST <A href="file:///E:/project/intro.html">GO
ON
TOUR</A> </FONT></B></P></FORM></BODY></HTML>
```

## HARRY PAGE SOURCE CODE

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<!-- saved from
url=(0064)http://www.randomhouse.com/catalog/display.pperl?isbn=080728
2596 --><HTML><HEAD><TITLE>Harry Potter and the Goblet of Fire by
J. K. Rowling, read by Jim Dale</TITLE>
<META content="text/html; charset=iso-8859-1" http-equiv=Content-
Type>
<META
content="Harry Potter and the Goblet of Fire J. K. Rowling, read by Jim
Dale Juvenile Listening Library Audio CD (Unabridged)"
name=description>
<META
content="&#10;Read by Jim Dale<br><br>Running time: 20 hrs., 30 mins.
17 CDs.<br><br>Harry Potter returns to Hogwarts for his fourth year of
magical adventures in Harry Potter and the Goblet of Fire. This year Harry
turns 14 and becomes interested in girls -- one in particular. And with Dark
Magic comes danger, as someone close to Harry dies. You'll have to listen
to learn more! The audio is available on July 8th. <br><br>"
name=keyword>
<META content=Catalog name=category>
<META content="MSHTML 5.00.2314.1000"
name=GENERATOR></HEAD>
<BODY aLink=#003366 bgColor=#ffffff link=#990000 vLink=#003366>
<TABLE border=0 cellPadding=0 cellSpacing=0 width=600>
  <TBODY>
  <TR>
    <TD align=right vAlign=top width=125><BR><BR><BR><BR>
      <CENTER><IMG src="harry_files/nakku.gif"> &nbsp;
</CENTER><BR><BR><BR><A
      href="http://www.randomhouse.com/catalog/display.pperl?isbn=0-8072-
8259-6#desc"><IMG
      alt="ABOUT THIS BOOK" border=0 height=17
src="harry_files/nav_desc.gif"
      width=125></A><BR><BR><A
      href="http://www.randomhouse.com/catalog/display.pperl?isbn=0-8072-
8259-6#bio"><IMG
```

```
    alt="ABOUT THE AUTHOR" border=0 height=17
src="harry_files/nav_bio.gif"
    width=125></A><BR><BR><BR><IMG height=100
    src="harry_files/felixthecat.gif" width=230>
    <P><A
href="http://www.randomhouse.com/features/grisham/">&nbsp;<IMG
    alt="John Grisham" border=0 height=210
src="harry_files/biggrisham2.gif"
    width=175></A></P>
    <P>&nbsp;</P></TD>
   <TD rowSpan=2 vAlign=top width=9><IMG alt="" height=1
    src="harry_files/space.gif" width=8> </TD>
   <TD bgColor=#000000 rowSpan=2 width=1><IMG height=1
    src="harry_files/space.gif" width=1> </TD>
   <TD rowSpan=2 vAlign=top width=10><IMG height=1
    src="harry_files/space.gif" width=10> </TD>
   <TD vAlign=top width=455><BR><BR><FONT color=#003366
size=+2>Harry Potter
    and the Goblet of Fire</FONT><BR><FONT color=#003366
    size=-1></FONT><BR><FONT color=#003366 size=-1><A

href="http://www.randomhouse.com/catalog/results.pperl?author=J%2e%20
K%2e%20Rowling&amp;author=Jim%20Dale">J.
    K. Rowling, read by Jim Dale</A> </FONT><BR><BR
clear=all><BR><IMG
    align=left alt=0-8072-8259-6 border=1 height=150 hspace=10
    src="harry_files/0-8072-8259-6.gif"

vspace=10>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<BR>&nbsp;
    <BR><BR><A name=desc><FONT color=#003366
    face="Trebuchet MS,Helvetica,Arial" size=2><B>ABOUT THIS
BOOK</B>
    </FONT><BR><BR>Read by Jim Dale<BR><BR>Running time: 20
hrs., 30 mins. 17
    CDs.<BR><BR>Harry Potter returns to Hogwarts for his fourth year of
    magical adventures in Harry Potter and the Goblet of Fire. This year
Harry
    turns 14 and becomes interested in girls -- one in particular. And with
    Dark Magic comes danger, as someone close to Harry dies. You'll have
to
    listen to learn more! The audio is available on July 8th.
    <BR><BR><BR><BR></A><A name=bio><FONT color=#003366
    face="Trebuchet MS,Helvetica,Arial" size=2><B>AUTHOR
BIOGRAPHY</B>
    </FONT><BR><BR>J.K. Rowling was a struggling single mother when
she wrote
    the beginning of Harry Potter and the Sorcerer's Stone, on scraps of
paper
    at a local cafe. But her efforts soon paid off, as she received an
    unprecedented award from the Scottish Arts Council enabling her to
finish
    the book. Since then, the debut novel has become an international
    phenomenon, garnering rave reviews and major awards, including the
British
    Book Awards Chidren's Book of the Year and the Smarties Prize. Ms.
Rowling
    lives in Edinburgh with her daughter.<BR><BR>Performer Bio: The
New York
    Times hailed Jim Dale as "The Toast of Broadway" in his title role in the
    musical Barnum. He has a long list of credits on the stage and in film
and
    was nominated for an Oscar for writing the lyrics for Georgy Girl.

<BR><BR>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
    </A>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; <A
    href="file:///E:/project/alphalist.html"><I><FONT face=Calligrapher
    size=4>next</FONT></I><BR><FONT
size=2><BR></FONT></A><FONT

size=2>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbs
```

```
p;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n
bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;
     </FONT><FONT face=Calligrapher size=4><I><FONT
color=#990000><A
     href="file:///E:/project/booklist.html">continue
   tour</A></FONT></I></FONT></TD></TR>
  <TR>
   <TD align=right vAlign=top><A href="file:///a:/homepage.html"><IMG
     height=48 src="harry_files/hummerhome.jpg" width=156></A></TD>
   <TD colSpan=2 vAlign=top
width=455></TD></TR></TBODY></TABLE></BODY></HTML>
```

## Sample Forms

# WELCOME TO BOOKWORLD.COM

*We are happy and thankful for you for visiting our sight. BOOKWORLD has great collection of books which you can purchase online .....*

LIST OF TOPICS

NOVELS | ENTERTAINMENT | TECHNICAL
SPORTS | POLITICS | SCIENCE
LITERATURE | ASTRONOMY | FICTION

We have the greatest search engine implemented here... The booglie search.

Search [ ] booglie

**JUST GO ON TOUR**

| | |
|---|---|
| User Id | Karthikr |
| Password | ******* |
| Name | Karthik |
| Email Id | karthikr9@usa.net |
| Address | 36, P.B.N Colony, |
| Phone | 98422 15336 |
| Credit Card Number | 11111 3643367 4123 |
| Company of the Credit Card | Amex |

submit reset cancel

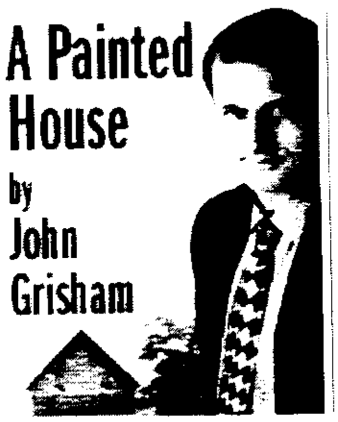
## Sample Output

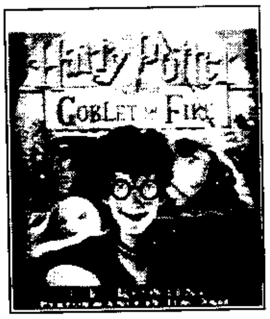
# Harry Potter and the Goblet of Fire

J. K. Rowling, read by Jim Dale

ABOUT THIS BOOK

ABOUT THE AUTHOR





ABOUT THIS BOOK

Read by Jim Dale

Running time: 20 hrs., 30 mins.
17 CDs.

Harry Potter returns to Hogwarts for his fourth year of magical adventures in Harry Potter and the Goblet of Fire. This year Harry turns 14 and becomes interested in girls -- one in particular. And with Dark Magic comes danger, as someone close to Harry dies. You'll have to listen to learn more! The audio is available on July 8th.

AUTHOR BIOGRAPHY

J.K. Rowling was a struggling single mother when she wrote the beginning of Harry Potter and the Sorcerer's Stone, on scraps of paper at a local cafe. But her efforts soon paid off, as she received an unprecedented award from the Scottish Arts Council enabling her to finish the book. Since then, the debut novel has become an international phenomenon, garnering rave reviews and major awards, including the British Book Awards Chidren's Book of the Year and the Smarties Prize. Ms. Rowling lives in Edinburgh with her daughter.

Performer Bio: The New York Times hailed Jim Dale as "The Toast of Broadway" in his title role in the musical Barnum. He has a long list of credits on the stage and in film and was nominated for an Oscar for writing the lyrics for Georgy Girl.

next

# Conclusion

## CONCLUSION

*It was a great experience making this e-com site "www.bookworld.com". The project has taught us a lot. It has also given us a lot of confidence, making us believe in ourselves.*

*We have tried to make the site very simple and user-friendly. It has given us a taste of the real world. Our sincere attempts in making a B2C website have been fruitful. The successful completion of the project has been a valuable reward for our arduous efforts.*

*We are highly indebted to the Head, the Staff and the Technical Staff of the Computer Science and Engineering Department for having given us the best of the facilities and their support. We dedicate this project to our Alma Mater, Kumaraguru College of Technology, for this where we have blossomed into consummate professionals.*

# Bibliography

# BIBLIOGRAPHY

- *The complete Reference – Java 2*

  *Patrick Naughton*

  *Herbert Schildt*

- *Servlet Programming*

  *O Reilly*

- *The internet*

*www.sun.java.com*

*www.fabmart.com*

- *Java Servlets and Server*

  *O Reilly*