

APPLICATION SERVERS

"BUILDING DYNAMIC DATABASE-DRIVEN WEB APPLICATIONS"



Prepared for:

Dr. James F. LaSalle
Management Information Systems
Eller College of Business and Public Administration
The University of Arizona

Prepared by:

Chris Ballam

Originally submitted December 13, 2001; republished August 20, 2026, for archival reference.

Contents

1	Introduction	4
1.1	What is an Application Server?	4
1.2	Application Server Overview	6
2	Current Application Servers	7
2.1	ColdFusion 5.....	7
2.2	IIS 5.0	8
2.3	JRUN 3.1	9
2.4	Tomcat 3.2	9
2.5	WebSphere 4.0.....	10
2.6	Which Application Server to Use?	11
3	Current Server-Side Scripting Languages	12
3.1	ASP 3.0	12
3.2	ColdFusion 5.....	13
3.3	JSP 1.2.....	14
3.4	PHP 4.0.....	14
3.5	Which Server-Side Scripting Language to Use?	15
4	Future of Server-Side Scripting Languages	16
4.1	ASP.NET.....	16
4.2	ColdFusion 6 (Neo)	17
4.3	JSP 1.3.....	18
4.4	PHP 4.1.....	19
5	The Design of My Web Application	21
5.1	Overview	21
5.2	Technologies Used	21
5.3	Database Design	22
5.4	Main Page	23
5.5	Add Movies	24
5.6	View Movies.....	27
5.7	Edit Movies	29
6	Summary	31
7	Web Application ASP Code	33
7.1	Add Movie (1 of 4) (AddMovie.asp)	33
7.2	Add Movie (2 of 4) (AddMovie2.asp)	39
7.3	Add Movie (3 of 4) (AddMovie3.asp)	43
7.4	Add Movie (4 of 4) Part A (AddMovie4.asp).....	47
7.5	Add Movie (4 of 4) Part B (AddMovie5.asp).....	50
7.6	Add an Actor (AddActor.asp).....	54
7.7	Edit a Movie (1 of 4) (EditMovie.asp)	57
7.8	Edit a Movie (2 of 4) (EditMovie2.asp)	59

7.9	Edit a Movie (3 of 4) (EditMovie3.asp)	65
7.10	Edit a Movie (4 of 4) (EditMovie4.asp)	67
7.11	View Movie (ViewMovie.asp)	68
7.12	View Movie Results (ViewMovieResults.asp).....	70
8	Back-End Database SQL Code (SQL Server).....	77
8.1	Create Tables.....	77
8.2	Insert Set Data.....	79
8.3	Insert Sample Movies	87
8.4	Insert Sample Borrowers	89
8.5	Drop All Tables	89
9	References	90

1 INTRODUCTION

1.1 What is an Application Server?

For the last three years the application server market has been completely driven by two companies, IBM and BEA Systems (Phillips, 2001). Today there are many different kinds of application servers available such as: Macromedia's ColdFusion Server, Microsoft's Internet Information Server (IIS), Macromedia's JRUN, Apache's Tomcat, and IBM's WebSphere to name a few. Each of the aforementioned application servers can perform the same basic tasks, but they each take advantage of different technologies and platforms. I will discuss the individual application servers in more detail later in this paper.

Typically, application servers work with Web servers to deliver dynamic Web pages. "Any Internet-based application that generates personalized or dynamic content" relies on application servers to make this happen (Dyck, 2001). Application servers provide a way for Web developers to add "intelligence" to their Web sites (Dyck, 2001). The reason why application servers add intelligence to Web sites is because they can take data that has been entered into a Web browser and they can "run custom business logic on the input (looking up customer information in a back-end database, for example)" and then the application server generates HTML and sends it back to the user's Web browser (Dyck, 2001).

Web applications can be written using server-side scripting languages such as: ASP, ColdFusion, JSP, or PHP or whatever server-side scripting language the application server being used supports. The application server recognizes the server-side scripts and executes them on the server. Executing server-side scripts could include tasks like calculating values or retrieving information from a back-end database. After the application server has executed the server-side scripts, it then generates HTML Web pages which it passes to the Web server to display on the Web. Application servers are referred to by some as "the brains behind a Web site" (Dyck, 2001).

Application servers are the future of dynamic database driven Web applications because they allow developers to create Web applications that are completely driven by back-end databases. Another huge advantage of using an application server is that allows developers to create and provide the business logic for Web applications (Sergey, 2001). Application servers also provide a “runtime environment and back-end services that give applications the stability, reliability, and performance most enterprises require” (Desai, 2001).

Application servers also provide the services and tools for generations of Web user interfaces, all the Web related protocols and infrastructures, and they support all the Web programming models (Stoeckel, 1999).

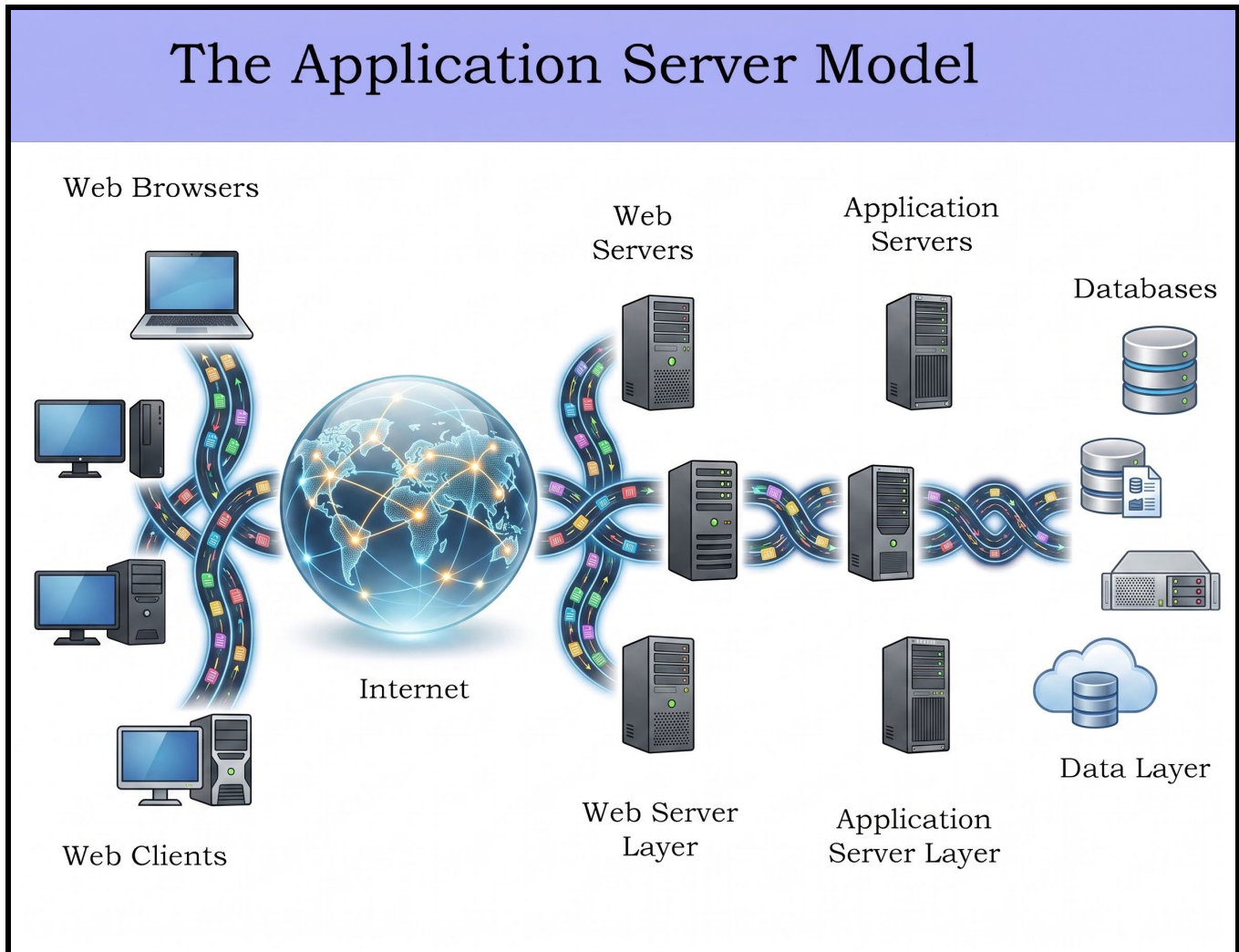
Because Web applications running on application servers can be accessed by anyone with Internet access, it’s hard to predict what kind of load you will have on your system so application servers also have built in load balancing techniques that will allow your Web application to withstand all the demands placed on it (Stoeckel, 1999).

Application servers also have certain security features built into them. Many of the application servers available today have some sort of encryption capabilities to protect the transmitted data (Stoeckel, 1999). Many application servers also support certain protocols like HTTP tunneling which allows you to set up your Web applications behind a firewall (Stoeckel, 1999).

As today’s companies are moving away from simple transaction e-commerce Web sites and building very complex and sophisticated e-commerce Web sites, application servers are the solution to providing developers with the tools they need to incorporate business logic in their e-commerce sites (Desai, 2001). Some even suggest that “today’s application server market and the e-commerce market are beginning to converge” (Desai, 2001).

1.2 Application Server Overview

Application servers are usually viewed as part of a three-tiered application as illustrated in the figure below:



The first tier is usually the front-end Web based graphical user interface. The first tier is served to the end user's Web browser by some sort of Web server. The middle-tier provides the business logic of the Web application. The middle-tier is served to the Web server by the application server. The third tier consists of a back-end database and transaction server which provides the application server with the data it requests from the database (Sergey, 2001).

Many of the Web sites being developed today need to be very reliable, scalable, stable, and manageable (Garris, 1998). Because Web sites today are being used as business-critical applications, a new model for Web development is emerging. Application servers make this new Web development

model a reality because they allow developers to incorporate business logic into their Web sites (Garris, 1998).

The application server acts as the middleman between a front-end Web site and the back-end database greatly increasing the power and extensibility of a Web application because the Web pages are generated when they are requested by the user's Web browser. The model of generating Web pages at the moment the user requests the document from their Web browser gets rid of the static Web page model and allows dynamic Web applications to be created on-the-fly (Desai, 2001).

2 CURRENT APPLICATION SERVERS

2.1 ColdFusion 5

ColdFusion 5 Server is Macromedia's proprietary Web application server. ColdFusion was first developed by Allaire Corporation but is now considered a Macromedia product because of the Allaire Macromedia merger in March 2001. ColdFusion 5 Server is available on Windows, Solaris, Red Hat Linux, and HP-UX platforms (Walker, 2001). ColdFusion 5 Server is also available in three different editions: Express, Professional, and Enterprise (Walker, 2001).

The Express Edition of ColdFusion is for personal use and can be downloaded for free and can be downloaded from Macromedia's Web site (Desai, 2001). The Professional Edition is designed for small-scale deployments and costs \$1,295 (Desai, 2001). And finally, the Enterprise Edition is designed for large scale deployment and costs \$4,995 (Desai, 2001).

ColdFusion 5 Server has a very easy-to-use development environment that doesn't require much prior programming experience (Dyck, 2001). With ColdFusion 5 Server, Macromedia is focusing on "midsize organizations that need robust tools to help them get up and running quickly with sophisticated applications, without exorbitant costs" (Desai, 2001).

ColdFusion 5 Server is based on a tag-based approach rather than based on a previously existing programming platform such as Java or Visual Basic (Desai, 2001). The ColdFusion 5 Server

development environment is also very customizable and is all visual with a browser based administrative tool (Desai, 2001).

Because ColdFusion 5 Server does not require much previous programming experience and the learning curve is very low, ColdFusion 5 Server is one of the most preferred application servers available today for midsize organizations (Desai, 2001).

2.2 IIS 5.0

Microsoft's Internet Information Server (IIS) 5.0 isn't a true application server. It is more like a Web server that can do the same things as an application server. Another unique thing about Microsoft's Internet Information Server 5.0 is that they don't sell it, they bundle it with their Windows Operating System (Lam, 2001).

Internet Information Server 5.0 recognizes and executes Web pages that are written using the server-side scripting language Active Server Pages (ASP). When IIS encounters the special ASP identifiers of <% and %> it executes the ASP code between the <% tags and then sends the result back to the user's Web browser as HTML. Because all the ASP code was executed only on the server, the end user will not see any of the underlying ASP code because the ASP code stays at the server end and is not transmitted to the user's browser. Internet Information Server 5.0 executes the underlying ASP code and then generates HTML and only sends the dynamically generated HTML to the end user. (Lam, 2001). Internet Information Server 5.0 will accept and serve both VBScript and JScript ASP pages (Walker 2001).

Microsoft's Internet Information Server 5.0 takes advantage of Component Object Model (COM+) objects which are kind of like Microsoft's version of Java 2 Platform, Enterprise Edition's (J2EE) Enterprise JavaBeans (EJB) components (Lam, 2001).

2.3 JRUN 3.1

Besides having a proprietary ColdFusion 5 Server, Macromedia also has another application server named JRUN 3.1. JRUN was also first developed by Allaire Corporation, but it is now considered a Macromedia product because of the Allaire Macromedia merger in March 2001.

JRUN 3.1 is based on the Java 2 Platform, Enterprise Edition (J2EE) platform and supports Web applications built using Java technology like server-side scripting languages such as JavaServer Pages (JSP) (Desai, 2001).

JRUN 3.1 is available in four different editions: Developer, Professional, Enterprise, and a Studio Edition (Desai, 2001). The Developer Edition of JRUN 3.1 is for personal use and can be downloaded for free from Macromedia's Web site (Desai, 2001). The Professional Edition is designed for small-scale deployments and costs \$795 per CPU (Desai, 2001). The Enterprise Edition is designed for large-scale deployment and costs \$4,995 per CPU (Desai, 2001). And finally, the JRUN 3.1 Studio Edition costs \$495 per seat (Desai, 2001).

Like ColdFusion 5 Server, JRUN 3.1 also has a proprietary browser-based administration console (Desai, 2001). The JRUN 3.1 Studio "simplifies building front-end interfaces in HTML, Wireless Markup Language (WML), JSP, and ASP" (Desai, 2001).

Currently JRUN 3.1 is not as popular as ColdFusion 5 Server, but it incorporates and supports Java technologies which are where Web applications seem to be going in the future so I would bet that JRUN will increase in its use and popularity in the near future.

2.4 Tomcat 3.2

Tomcat 3.2 is Apache's current application server. Tomcat 3.2 is open-source, completely free, and can be downloaded from Apache's Web site (Dragan, 2001). The Tomcat 3.2 application server is based on the Java 2 Platform, Enterprise Edition (J2EE) platform and supports Web applications built using Java technology like server-side scripting languages such as JavaServer Pages (JSP) (Dragan, 2001).

The Tomcat application server platform is currently being developed by Apache and Sun Microsystems. Tomcat 3.2 also “easily hooks into Apache, Microsoft IIS, and Netscape Enterprise Web servers” (Dragan, 2001).

All the reviews I read about Apache’s Tomcat 3.2 and my personal experience showed that Tomcat 3.2 is very hard to configure and use. Tomcat 3.2 does not have any built-in development tools like most other application servers which means you are stuck using the command line Java compiler (Dragan, 2001). Besides requiring you to build your Web applications in the command line Java compiler, Tomcat 3.2 also requires you to know a great deal about the Java Virtual Machine (JVM) because there is no command console or browser-based administration module like there is in most of the other application servers available today (Dragan, 2001).

If the cost of an application server is a big issue, then Tomcat 3.2 is probably the best application server for you, but if you want an application server that is easy to use and has a low learning curve, then Tomcat 3.2 is not the right solution.

2.5 WebSphere 4.0

WebSphere 4.0 is IBM’s current application server. WebSphere takes advantage of Java technology and supports Web applications built using the JSP server-side scripting language. IBM’s target markets for WebSphere 4.0 are “large to midsize companies seeking an enterprise-level e-business solution and for those that want to develop in Java across a common platform” (Desai, 2001).

On a more technical side, WebSphere 4.0 uses “Java Database Connectivity (JDBC) to access DB2, Informix, Oracle and Sybase databases. It supports Common Object Request Broker Architecture (CORBA) 2.3, JavaBeans 1.0, and Enterprise JavaBeans (EJB) 1.0 object technologies, and includes an Object Request Broker (ORB), a Java 2 Platform, Enterprise Edition (J2EE) container and tools for wrapping Component Object Model (COM) objects” (Desai, 2001). WebSphere 4.0 also has “an array of speed-enhancing caching techniques, rock-solid load balancing and failover handling, and a complete complement of developer tools for creating Web applications” (Dyck, 2001).

Part of the reason why WebSphere 4.0 is the preferred application server for large companies is because it is very reliable and secure. The reliability and security of WebSphere do not come at a small price because the WebSphere Application Server Advanced Edition 4.0 costs \$7,500 per CPU license (Gunnerson, 2001).

2.6 Which Application Server to Use?

Because there are so many different types of application servers based on all kinds of technologies and protocols, choosing what type of application server to invest in is not an easy task. Most all the application servers I reviewed above would probably be able to support your Web application in one way or another; it just depends on what requirements and existing technologies you already have in your company.

If your company uses all Microsoft products, then your best solution might be to use Microsoft's Internet Information Server 5.0 because you already have it installed and you won't have to pay any additional fees to use it. Also, you will probably already have many developers that are familiar with Microsoft's COM+ objects which are supported and used by Active Server Pages. Already having developers familiar with Microsoft's COM+ objects, VBScript, or JScript is a huge reason to use Microsoft's Internet Information Server 5.0 because it is based on those technologies.

If your company does a lot of development using Java technologies, then you will probably want to use either Macromedia's JRUN 3.1, IBM's WebSphere 4.0, or Apache's Tomcat 3.2 because they are also based on Java technologies. Remember that JRUN 3.1 is designed for midsized companies, WebSphere 4.0 is designed for large and midsized companies, and Tomcat 3.2 isn't really designed for midsized or large companies. So, if you want a solution for a midsized company and you don't want to spend a fortune on an application server then JRUN 3.1 is probably the best choice for you.

If you don't have a budget for an application server and you are a Java Virtual Machine Wizard, then Tomcat 3.2 is probably the best choice for you because it is free and open-sourced. If you are looking for an application server for a large sized company and money is not an object, WebSphere 4.0

is probably the best solution because it will be able to handle all the demands of a large company and it is very secure and reliable.

And finally, if you are looking for an application server that is very easy to use and has a very low learning curve, ColdFusion 5 Server is probably the application server for you. Many people prefer ColdFusion 5 Server because it doesn't really require any previous programming experience and it has a very easy to use graphical user interface. Many people prefer ColdFusion because it is probably the "quickest" way to build Web applications.

As you can see, determining which application server is right for you really depends on what you are going to be doing with your Web application, what technologies you already have in your company, and what type of programming experience and preference your developers have. Keeping all these things in mind will help you make the best choice of which application server you should use to create dynamic Web applications.

3 CURRENT SERVER-SIDE SCRIPTING LANGUAGES

3.1 ASP 3.0

Active Server Pages (ASP) is a technology that was developed by Microsoft as an extension to its line of web server software which includes Internet Information Server (IIS) and Personal Web Server (PWS) (Walker, 2001). ASP comes along with Microsoft's Internet Information Server and Personal Web Server at no extra cost, so if you have a current windows operating system, you also have the capability to use ASP (Walker, 2001).

ASP is used with HTML to create dynamic Web pages. The ASP code is written throughout HTML code inside of special `<% %>` tags. When the application server encounters the `<% %>` tags it executes the ASP code and then sends the result to the user's Web browser as HTML. The user cannot see the underlying ASP code because ASP is a server-side scripting language which means that all the ASP code stays at the server end and is not transmitted to the user's browser. If the user tries to view

the source of the ASP page, they will only see the dynamically generated HTML from the application server.

ASP Web pages typically have .asp as the file extension. ASP also supports two different scripting languages: JScript and VBScript. ASP also has the capability of supporting other scripting languages if configured correctly (Walker, 2001). JScript is kind of like Microsoft's own version of JavaScript (Lam, 2001). VBScript is a subset of Microsoft's Visual Basic language which means that ASP pages that use VBScript can take advantage of ActiveX controls, automation servers, and Java applets (Walker, 2001).

3.2 ColdFusion 5

ColdFusion uses CFML which stands for ColdFusion Markup Language as its server-side scripting language. Like the ColdFusion 5 Server, CFML is also proprietary and was originally created by Allaire Corporation. CFML and ColdFusion are now owned by Macromedia, Inc. due to the Allaire Macromedia merger earlier this year.

The ColdFusion Markup Language can take advantage of many available Internet technologies, including Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Hypertext Transfer Protocol (HTTP), Lightweight Directory Access Protocol (LDAP), File Transfer Protocol (FTP), Extensible Markup Language (XML), text search engines, and file servers (Walker, 2001). ColdFusion is also extensible because it can take advantage of the following architectures and models: Common Object Request Broker Architecture (CORBA), Component Object Model (COM), Java servlets, and JavaBeans (Walker, 2001). ColdFusion pages usually have a .cfm extension that the ColdFusion server recognizes.

Another really nice thing about ColdFusion is that it comes with a really nice Integrated Development Environment (IDE) to help you create your ColdFusion Web applications. The built in IDE is all visual and much of the ColdFusion code is generated for you while using the IDE.

3.3 JSP 1.2

JavaServer Pages (JSP) is a Java-based server-side scripting language that is part of Sun Microsystems's Java Technology (Walker, 2001). JSP allows you to create dynamic Web pages by using XML tags and scriptlets. Scriptlets are programs that have been written in Java. JSP was created to extend Java servlet technology. Servlets are "platform-independent server-side modules that are used to extend a Web server's functionality by compiling the pages into servlets" (Walker, 2001).

JSP pages usually have a .jsp extension that the JSP application server recognizes. One important thing about JSP pages is that they are only compiled the first time they are requested from a Web browser. After the first request, the compiled JSP page is cached by the servlets engine in memory for further requests. This caching of the compiled JSP pages really speeds up subsequent requests of the pages (Walker, 2001).

Another huge advantage of the JSP server model is that it can take advantage of many of the features of Java including integration with JavaBeans and all the API's that Java has to offer (Walker, 2001).

3.4 PHP 4.0

Like all the previous server models mentioned above, PHP is also a server-side scripting language used to generate content for the Web. PHP gets its name from the first version of PHP which was named "Personal Home Page Tool" (Culbert, 2001). Most PHP pages end with the extension .php, .php3, or .phtml. PHP can be used to "query databases, generate dynamic graphs, output PDFs on the fly, create certain types of images on the fly, and parse XML" (Culbert, 2001).

One unique and distinguishing thing about PHP was that it was designed specifically for the Web (Culbert, 2001). Other languages like Java and VBScript weren't originally designed for the Web, but they have been modified to allow for Web content creation. Another unique thing about PHP is that it is free and Open Source which really appeals to the Linux and Open-Source developers and users (Jepson, 2001).

Because Linux is one of the “most popular platforms for serving Web content,” PHP is a very good open source solution because it is very similar to the UNIX based script programming language Perl (Practical Extraction and Reporting Language) which is widely used by Linux Web developers (Trachtenberg, 1999).

3.5 Which Server-Side Scripting Language to Use?

Like choosing the right application server, choosing the right server-side scripting language is no easy task. With so many different types of server-side scripting languages available today, it is hard to determine which one is the best fit for your company.

If you are looking for a “Microsoft Solution” then the server-side scripting language for you is definitely ASP. ASP is probably the best choice if your Web developers are already familiar with VBScript or JScript because ASP can accommodate either scripting language. ASP is also a good choice if you are already using a Windows 2000 Server because IIS 5.0 comes bundled with Windows 2000 Server which means you won’t have any additional costs associated with using ASP as your server-side scripting language.

If you want to take advantage of Java technologies, your best choice is to use JSP because it is based on Java and can do just about everything Java can do. If the cost of the application server is an issue, then JSP could still work in your company if you are willing to use the open-source application server Tomcat 3.2. Using Tomcat 3.2 and JSP will not cost your company anything, but it may be harder to get working than other JSP solutions like JRUN 3.1 or WebSphere 4.0.

If money is not a big problem, then you could use JSP with JRUN 3.1 or WebSphere 4.0. Using JSP and JRUN 3.1 is a good solution for a midsized company, but if you are looking for a solution in a large company then you will probably want to use JSP and WebSphere 4.0.

Another alternative is to use CFML and the ColdFusion 5 Server. One of the great things about using CFML is that it doesn’t really require someone with a lot of programming experience because it is tag-based and very easy to learn. The ColdFusion solution is not an inexpensive one because you will

have to pay about \$5,000 for the ColdFusion 5 Server. ColdFusion is also designed for midsized companies.

And finally, if you are a Linux buff and you love open-sourced solutions then PHP is definitely the server-side scripting language for you. PHP is especially the right solution if your Web developers already use Perl because PHP was developed with Perl in mind and they are very similar in syntax. PHP also won't cost you anything to implement because everything you need is free.

Overall, the best thing to keep in mind when trying to determine what server-side scripting language to use is what kinds of technologies do I already have, what technologies do I already use, and what kinds of technologies do I already know. If you know VBScript, then choose ASP. If you know Java, then choose JSP. If you know Perl, choose PHP. If you don't have much programming experience under your belt and you are looking for a simple and quick solution, then ColdFusion's CFML is probably the best solution for you.

4 FUTURE OF SERVER-SIDE SCRIPTING LANGUAGES

4.1 ASP.NET

ASP.NET is Microsoft's upgrade from ASP 3.0. Microsoft estimates ASP.NET will be officially released in 2002 along with the .NET Server. Like ASP 3.0, ASP.NET is a server-side scripting language.

One major difference between ASP 3.0 and ASP.NET is that ASP 3.0 only supports interpreted VBScript and JScript but ASP.NET supports "more than 25 .NET languages (including built-in support for VB.NET, C#, and JScript.NET – no tool required), giving you unprecedented flexibility in your choice of language" (Why ASP.NET?, 2001).

ASP.NET development can be done by using any text editor or you can also use Visual Studio .NET which is an environment which allows you to create ASP.NET Web applications in a visual manner. Visual Studio .NET will also be released in 2002 around the same time the .NET Server is

released (Desai, 2001). Visual Studio .NET will allow ASP.NET developers to use “UML class modeling, database modeling (conceptual, logical, and physical models), testing tools (functional, performance and scalability), and enterprise frameworks and templates” (Why ASP.NET?, 2001).

Another improvement in ASP.NET is better performance. ASP.NET will be much faster than the current ASP 3.0. Microsoft claims that most Web applications using ASP.NET will be 3 to 5 times faster than Web applications written using ASP 3.0 (Why ASP.NET?, 2001).

Microsoft is also claiming that ASP.NET will outperform J2EE (Java 2 Platform, Enterprise Edition). On Microsoft’s ASP.NET Web site they have a “head-to-head comparison of performance and scalability between Sun’s Java Pet Store J2EE blueprint application and the ASP.NET implementation” (Why ASP.NET?, 2001). Microsoft boasts that the ASP.NET implementation of the Pet Store “significantly outperformed” the JSP application claiming their implementation “required 1/4th as many lines of code, was 28 times faster (that’s 2700%), and supported 7.6 times as many concurrent users as J2EE, with only 1/6th as much processor utilization” (Why ASP.NET?, 2001). With all this pre-release hype, it will be interesting to see if Microsoft’s final version of ASP.NET can really live up to everything.

4.2 ColdFusion 6 (Neo)

On October 22, 2001, Macromedia announced the details of their next major release of Macromedia ColdFusion (Press Release, 2001). Macromedia ColdFusion 6 is being codenamed “Neo” during the development process. ColdFusion 6 incorporates leading Java based application server technology which means that ColdFusion 6 will run on any JSP application server such as IBM’s WebSphere, Apache’s Tomcat, or Macromedia’s own JRUN (Neo FAQ, 2001).

Macromedia also released that ColdFusion 6 will “support XML Web Services, interoperate with Microsoft .NET technologies, and integrate with other Macromedia products such as Macromedia Flash and Dreamweaver” (Press Release, 2001).

The release of ColdFusion 6 is a huge step for Macromedia because currently you must have a proprietary ColdFusion Server to serve ColdFusion CFML (ColdFusion Markup Language) pages and Web Applications. With the next release of ColdFusion, you will be able to serve ColdFusion CFML pages and Web applications using any kind of JSP application servers. The reason why this is such a huge improvement on the product is because ColdFusion Server is fairly expensive at \$1,295 for the Professional Edition and \$4,995 for the Enterprise Edition (Neo FAQ, 2001). This has the potential to save companies lots of money if they choose JSP application servers like Apache's Tomcat which is free.

Supporting JSP application servers also makes ColdFusion 6 much more extensible and gives companies a lot more choices to choose from. Adding additional application server choices greatly increases the appeal of ColdFusion because many companies may already have JSP application servers that will now serve up ColdFusion 6 CFML Web applications.

I see Macromedia's ColdFusion 6 a huge step in the right direction. ColdFusion already holds a large portion of the server-side scripting language market, but opening their proprietary CFML to run on JSP application servers will help them increase their control of the market.

4.3 JSP 1.3

The details of the next version of JSP haven't been officially released, but Sun is giving some clues about some of the features they are trying to implement in JSP 1.3. Sun is tentatively saying that JSP 1.3 will be released in the second half of 2002 in correlation to the release of J2EE (Java 2 Platform, Enterprise Edition) 1.4. There seems to be two main things that are going to be implemented in the next version of JSP 1.3.

The first addition to JSP 1.2 is a built-in expression language (Pelegrí-Llopart, 2001). Sun isn't specifying exactly what a built-in expression language could mean for JSP, but the idea behind it is to allow users to do things like "param:a.b.c" directly in JSP code (Pelegrí-Llopart, 2001).

The other addition Sun is hinting will be part of JSP 1.3 has to do with “defining JSP custom actions using JSP itself” (Pelegri-Llopart, 2001). Macromedia is also contributing to the development of JSP 1.3 specification by offering ideas on how to write tags from JSP pages like they do in JRUN (Pelegri-Llopart, 2001).

It will be interesting to see what ways Sun improves JSP 1.3 to compete with all the hype of ASP.NET. It looks like Sun will try and incorporate ASP.NET Web Services support in JSP 1.3 to stay abreast with ASP.NET (Pelegri-Llopart, 2001).

4.4 PHP 4.1

The next version of PHP, which will be PHP 4.1, also has some very interesting additions and improvements. The improvement that is getting the most attention is the improved security features. PHP 4.1 will have a new input interface that will greatly improve PHP’s security (Clowes, 2001). The new PHP 4.1 input interface will also be backwards compatible so all the previous input mechanisms and interface models from previous PHP versions will also work in the new input interface (Clowes, 2001).

PHP 4.1 will also have “highly improved performance in general” (Clowes, 2001). Along the same lines, PHP 4.1 will have major improvements in performance and stability under Windows (Clowes, 2001). This improved stability under Windows will solve many of the complaints of PHP not “fully working” on the Windows platform (Culbert, 2001).

In fact, PHP developers are boasting that PHP 4.1 is 30 times faster on the Windows platform (Clowes, 2001). Another major addition to PHP 4.0 is lots of fixes and new functions. Because PHP is built around customized functions for the Web, additional functions will greatly improve the power and extensibility of PHP (Culbert, 2001)

Making PHP 4.1 more stable on the Windows platform and increasing the overall performance and security will continue to promote PHP as a stable server-side scripting language capable of building dynamic Web applications.

5 THE DESIGN OF MY WEB APPLICATION

5.1 Overview

I decided that the best way for me to get a feel for application servers, server-side scripting languages, and building dynamic database-driven Web applications was to actually build my own Web application. I tried to find a project that would give me exposure to some real-world problems and solutions so I created a Web application like something you might find at Amazon.com.

Instead of building a Web application that incorporates everything from books to electronics, I narrowed the scope of my Web application to movies. My Web application keeps track of all the movies I own, allows me to add additional movies as I acquire them, allows me to edit existing movie profiles, allows me to delete movies, and it allows me to keep track of my movies if someone borrows them.

All the drop-down menus throughout my Web application are dynamically populated from data stored in the backend database. All the images that are displayed are also dynamically displayed on the user's browser from the image names stored in the backend database. Just about every aspect of my Web application is dynamically driven by the back-end database.

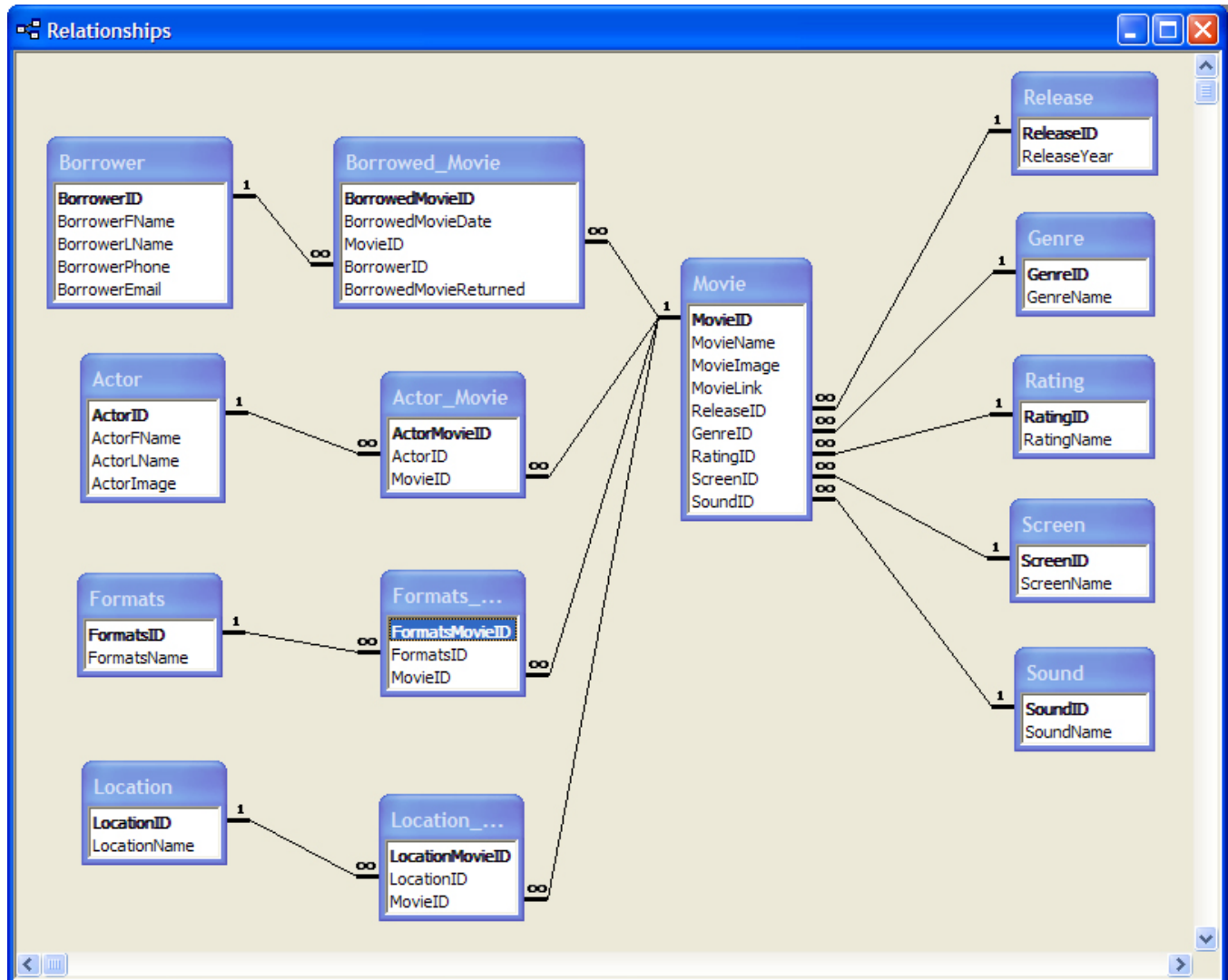
5.2 Technologies Used

I tried to use as many different technologies as I could on this Web application. I first built my Web application using the JSP server-side scripting language. I tried using both WebSphere and JRUN as JSP application servers. I used Apache as my Web server, and I connected to a back-end Oracle database using JDBC drivers.

After building my Web application using all Java tools and Oracle, I decided to build it again using all Microsoft tools. I used the ASP server-side scripting language, Internet Information Server (IIS) as both my application server and my Web server, and I connected to a back-end SQL Server database using ODBC drivers.

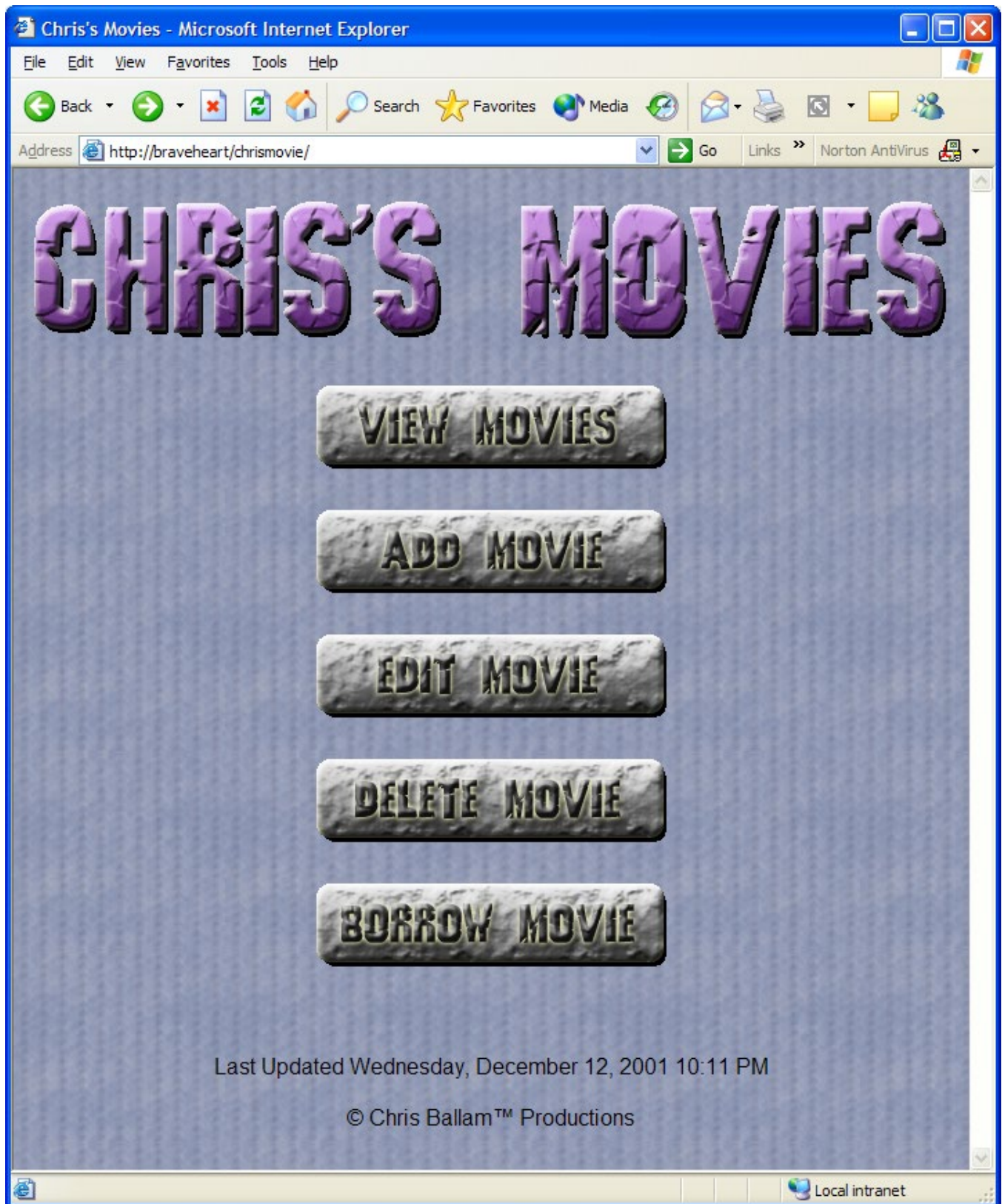
5.3 Database Design

I designed my database to allow for new movie formats, movie locations, actors, borrowers, release years, movie genres, movie ratings, screen formats, and sound formats. The diagram of my database's relational schema is shown below:



5.4 Main Page

Below is the first page of my Web application. This first page allows the user to view all the movies in the database, add a new movie, edit an existing movie, delete a movie, or borrow a movie.



5.5 Add Movies

Below are some screen shots of my Web application's method of adding new movies to the back-end database. I decided to break the add movie pages into four Web pages instead of having one huge Web page that requests all the information about the movie.

Add Movie (1 of 4) - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Reload Home Search Favorites Media Norton AntiVirus

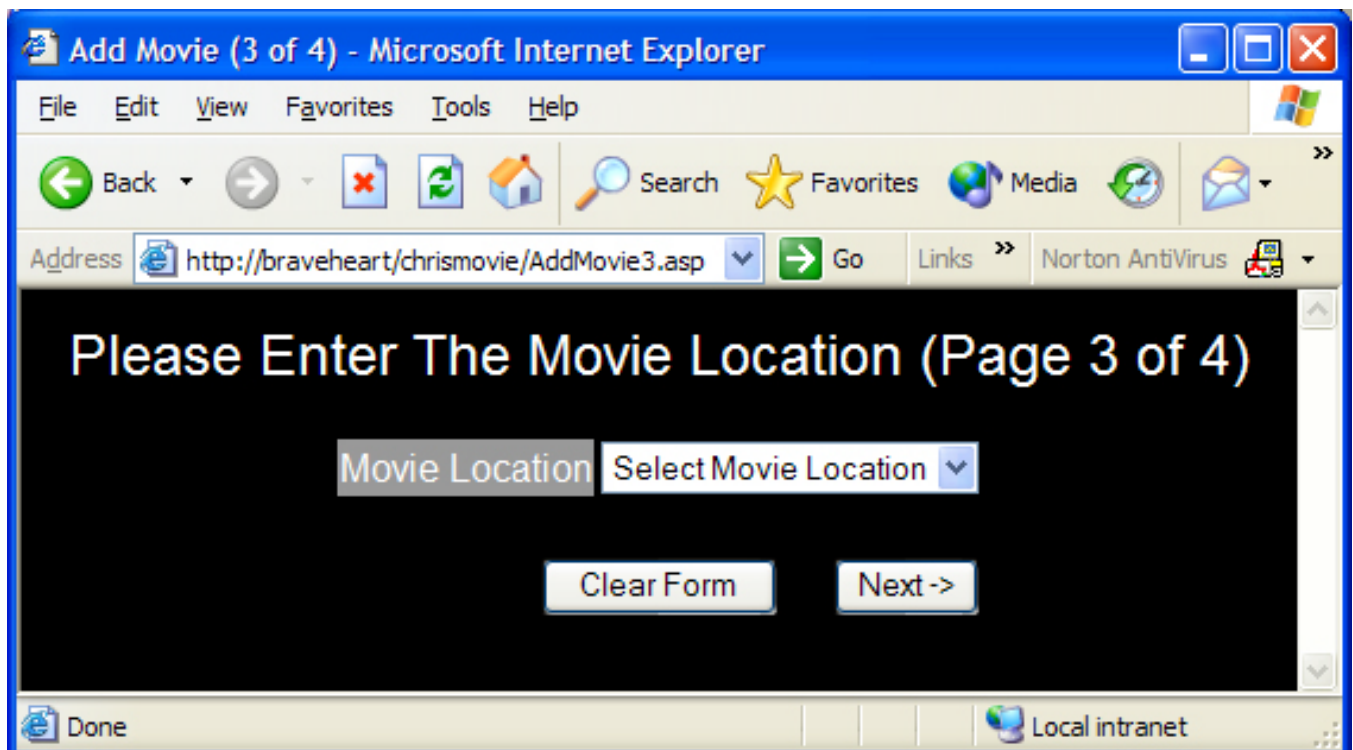
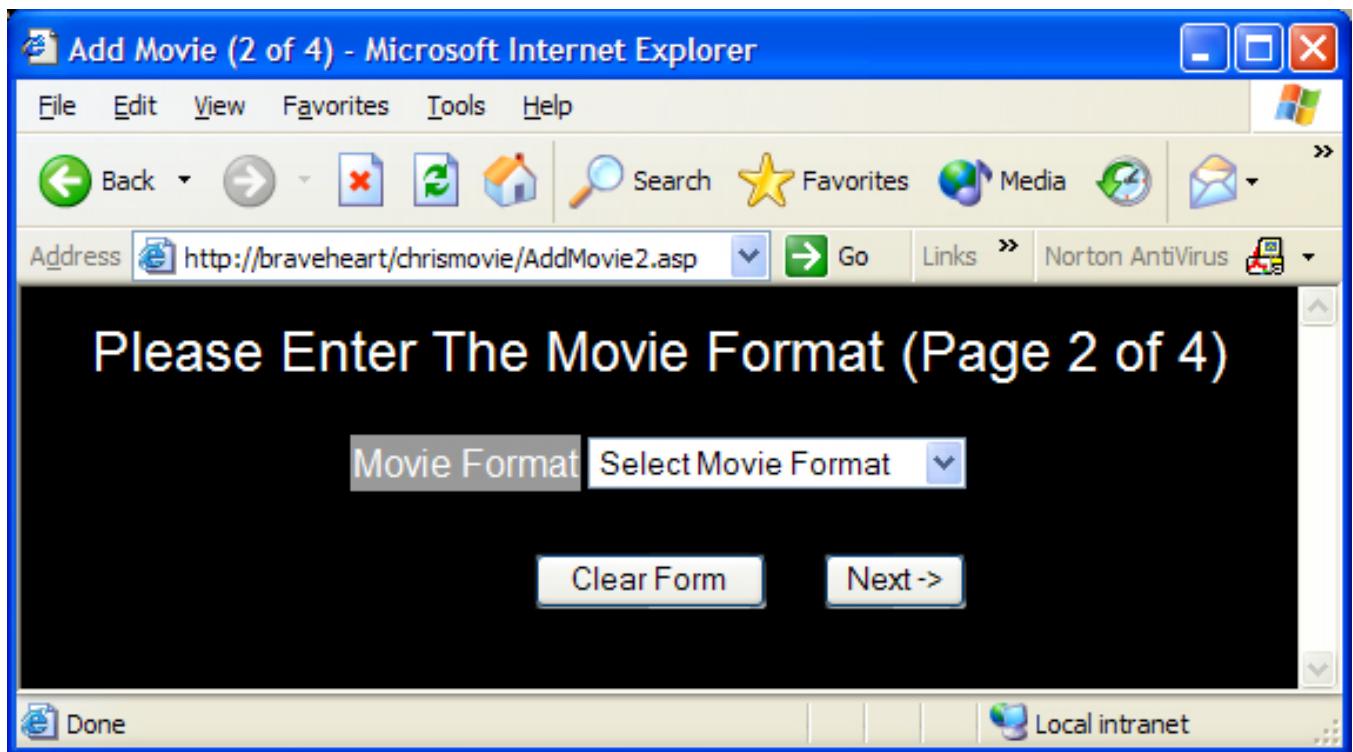
Address <http://braveheart/chrismovie/AddMovie.asp> Go Links Norton AntiVirus

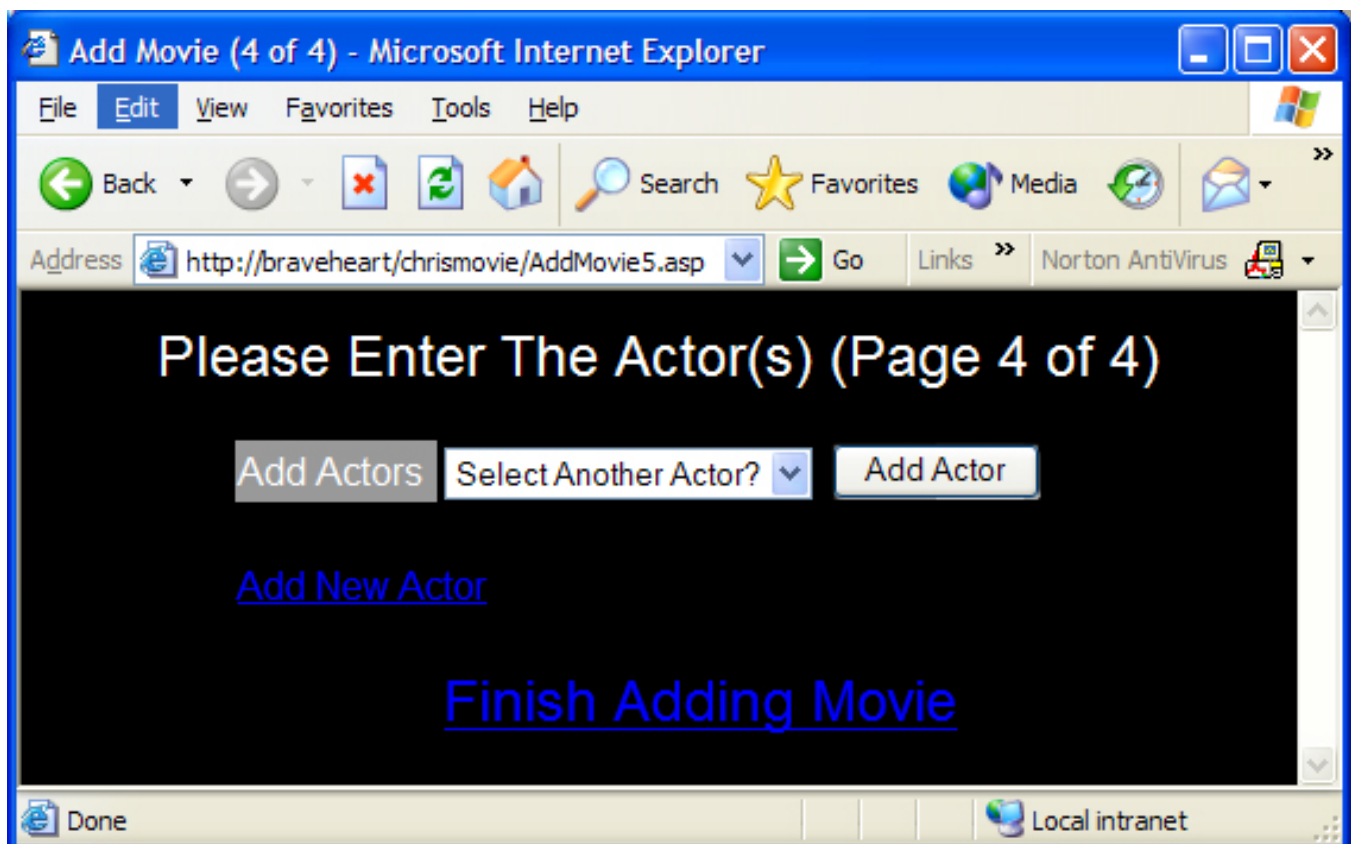
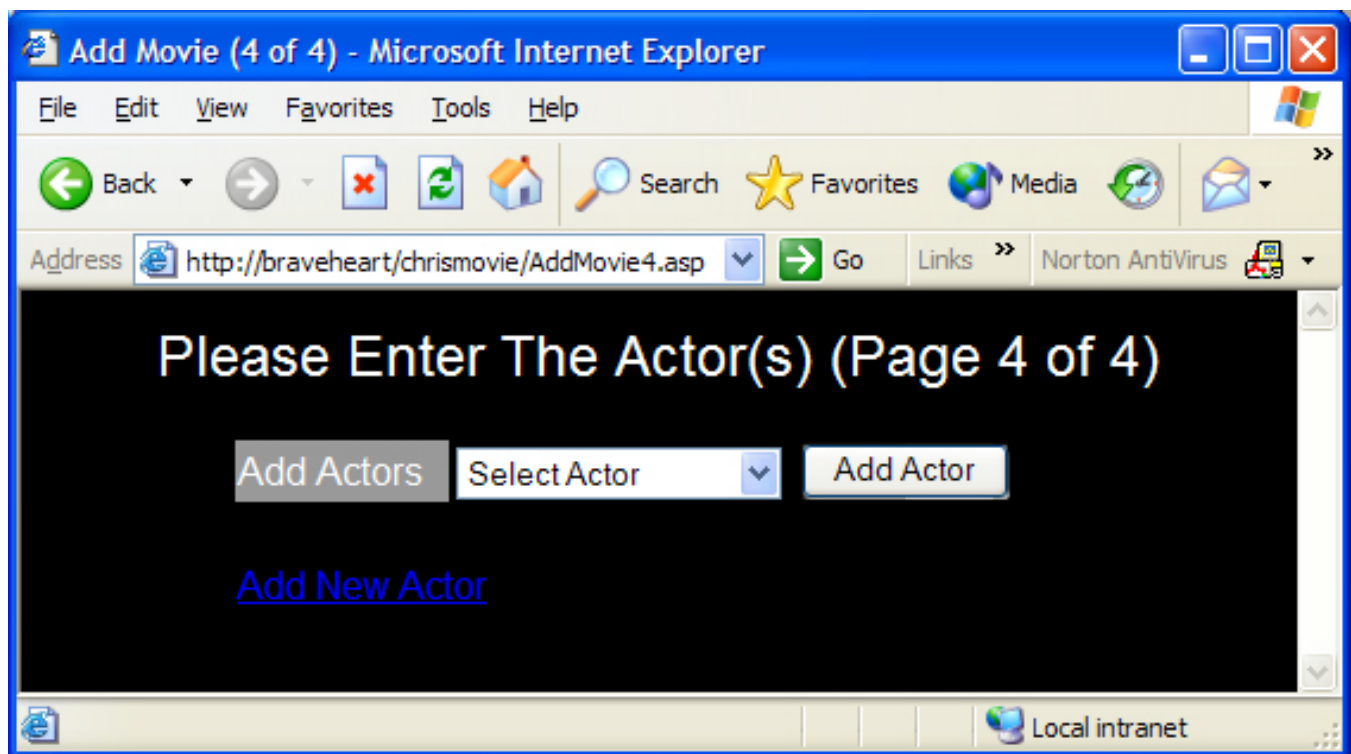
Please Enter The Movie Info (Page 1 of 4)

Movie Name	
Movie Image	
Movie Link	
Release Year	Select Year ▼
Genre	Select Genre ▼
Rating	Select Rating ▼
Screen Format	Select Screen Format ▼
Sound Format	Select Sound Format ▼

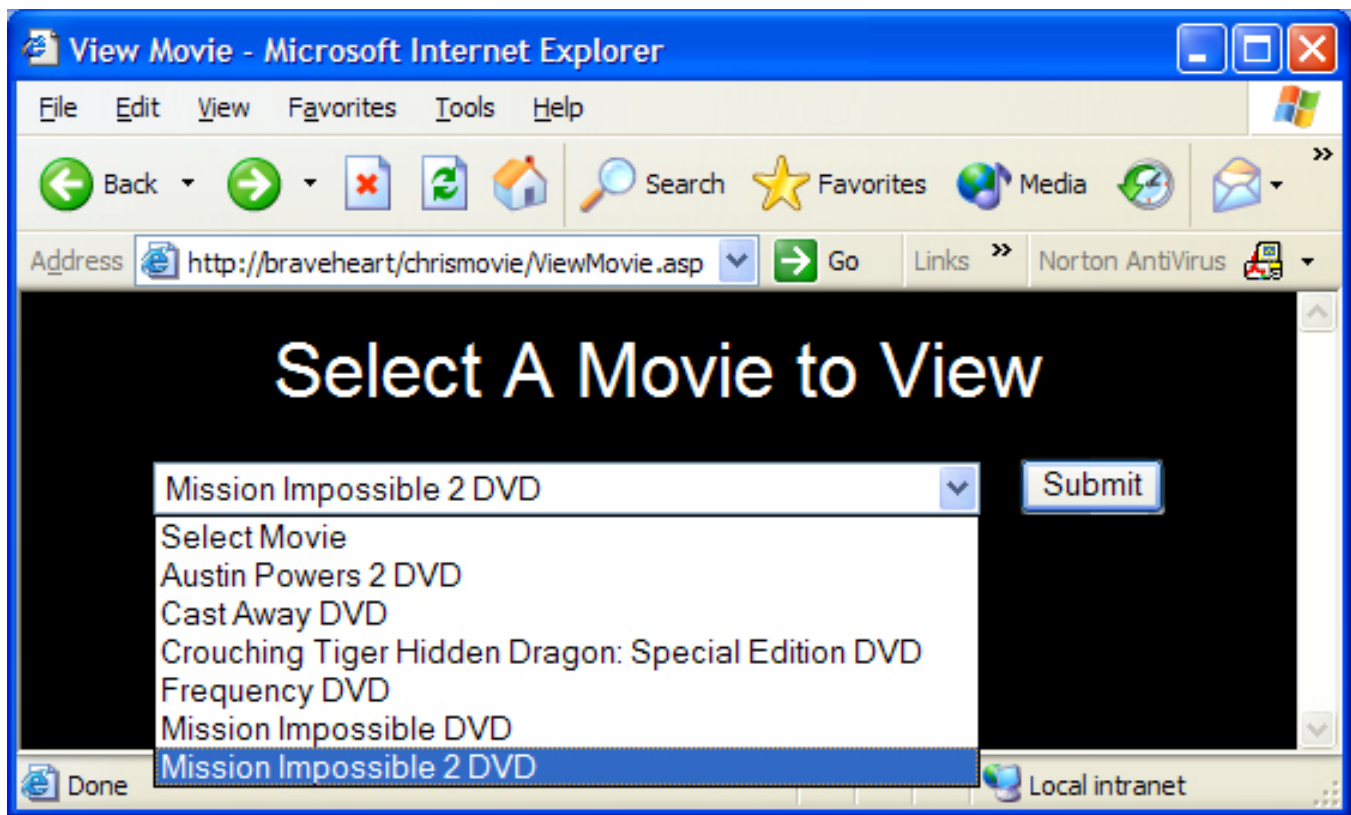
Clear Form Next ->

Done Local intranet





5.6 View Movies



After selecting “Mission Impossible 2 DVD” from the above screen, the following information is displayed about the Mission Impossible 2 DVD: The movie title, an image of the Movie Cover with an embedded hyperlink to the Internet Movie Database’s Web page on Mission Impossible 2, an external hyperlink to the Internet Movie Database’s Web page on Mission Impossible 2, the movie’s release year, genre, rating, screen format, sound format, movie format, movie location, and some of the star actors in the movie along with their images with imbedded hyperlinks to the Internet Movie Database’s Web page about them.

View Movie Results - Microsoft Internet Explorer

FileEditViewFavoritesToolsHelp

Back

Search

Favorites

Media

Go

Links

Norton Antivirus

Addresshttp://braveheart/chrismovie/ViewMovieResults.asp?Movie!

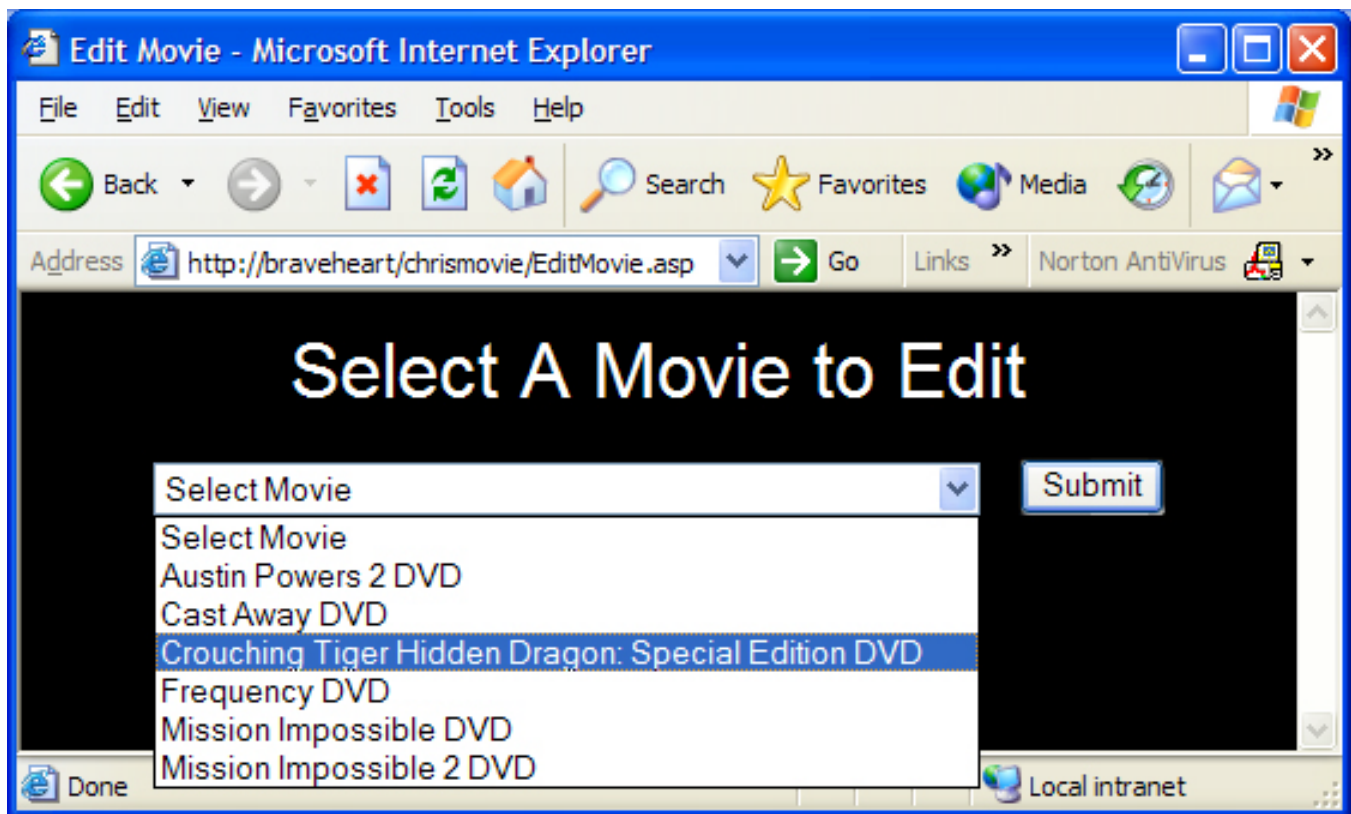
Movie Results

Movie Name	Mission Impossible 2
Movie Cover	
Movie Link	Mission Impossible 2 on IMDB
ReleaseYear	2000
Genre	Action / Adventure
Rating	PG-13
Screen Format	LBX / 16:9
Sound Format	5.1
Movie Format	DVD
Movie Location	DVD
Actors	Tom Cruise Douggray Scott Thandie Newton

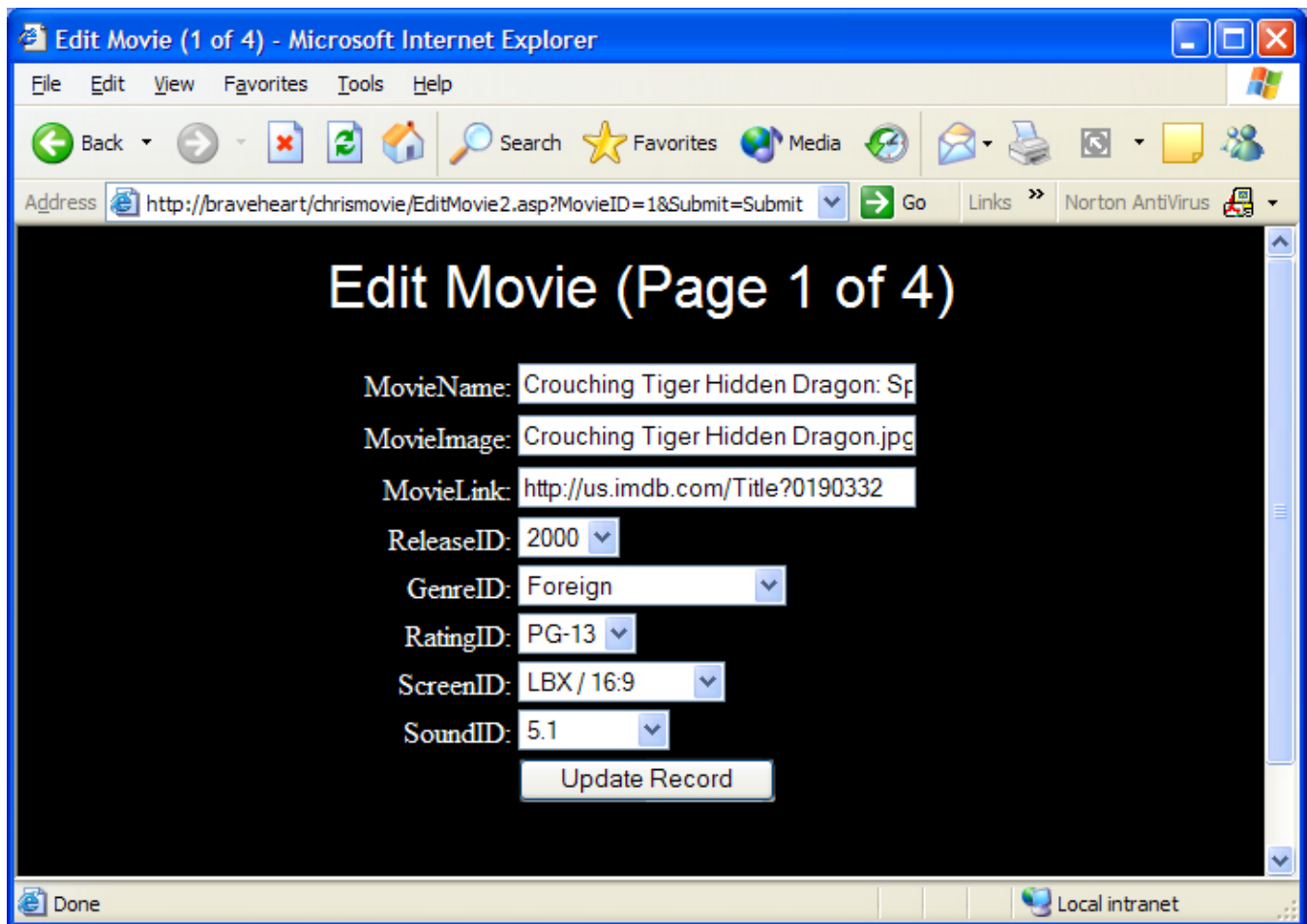
Local intranet

5.7 Edit Movies

Below is a screen shot of the page asking the user to select which movie they want to edit. The drop-down menu is dynamically populated by two different tables from the back-end database. Once a selection is made from the drop-down menu, the “Submit” button passes the Movie’s Unique ID number to the next page using the POST method.



In the above example, Crouching Tiger Hidden Dragon: Special Edition’s unique ID has been passed to the next page, the screen shot on the next page displays information about Crouching Tiger Hidden Dragon: Special Edition and allows users to make changes to the existing data in the back-end database.



6 SUMMARY

Although I was able to create my Web application using both JSP and ASP, I found that developing Web applications under each of these server-side scripting models can be very different. The first main difference I found was that JSP is fairly new and there isn't as much support and information out there right now. ASP, on the other hand, had lots of support and information available to me when I ran into problems. I also found that there are a lot more ASP developers out there than there are JSP developers. I found hundreds of very helpful Web sites devoted to ASP Web application development, but I could only find a handful of useful JSP Web development Web sites.

I also ran into many problems connecting my JSP Web application to a back-end Oracle database. It took me a long time to get the right JDBC drivers configured and working between my JSP Web application and my back-end database. I didn't have any major problems connecting my ASP Web application to a back-end database because the ODBC drivers I needed were already built into the Windows Operating System.

Overall, I found that it was much easier to develop my Web application using ASP instead of JSP. I didn't run into nearly as many problems getting the different technologies to communicate with each other and I found a lot more support for ASP.

After building this Web application and entering some movies into it I have already realized something that I will need to change for my Web application to handle many movies in the future. One major design change that I will need to make is to change the drop-down menus on the view movie, add movie, edit movie, and delete movie Web pages. Right now, because there are not very many movies entered into the database, a drop-down menu is sufficient but in the future as many more movies are entered into the database a drop-down menu is not sufficient because it will be difficult to scroll through hundreds of movies on a single drop-down menu. Instead of using drop down menus to display the movies I instead create a template that will show ten or twenty movies per page and have the option to

scroll through other templates of movies in the database. This will solve the usability issue of very long drop-down menus.

Another thing I have learned from creating this Web application is that your database design needs to be done very carefully and it needs to be able to support a dynamic Web site. Originally, I built my movie database for regular database use, but that design did not work when I tried to use it as a back-end database for my Web application. I had to completely change some of the tables in the database, so they supported things like image names and hyperlinks.

Overall, building this Web application has exposed me to dozens of technologies, protocols, application servers, and server-side scripting languages. I think the hands-on experience of creating my Web application far exceeds doing research on related topics because I got to see first-hand what the technologies were like and the associated advantages and disadvantages of using them. I also was able to create an entirely Web based Web application that will help me keep track of all my movies. And who knows? Maybe I will be able to improve my Web application enough so that I will be able sell it for millions of dollars.

7 WEB APPLICATION ASP CODE

7.1 Add Movie (1 of 4) (AddMovie.asp)

```
<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
// *** Edit Operations: declare variables

// set the form action variable
var MM_editAction = Request.ServerVariables("URL");
if (Request.QueryString) {
    MM_editAction += "?" + Request.QueryString;
}

// boolean to abort record edit
var MM_abortEdit = false;

// query string to execute
var MM_editQuery = "";
%>
<%
// *** Insert Record: set variables

if (String(Request("MM_insert")) != "undefined") {

    var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
    var MM_editTable = "dbo.Movie";
    var MM_editRedirectUrl = "AddMovie2.asp";
    var MM_fieldsStr =
"MovieNameText|value|MovieImageText|value|MovieLinkText|value|ReleaseSelect|value|GenreSelect|v
alue|RatingSelect|value|ScreenSelect|value|SoundSelect|value";
    var MM_columnsStr =
"MovieName|',none,'|MovieImage|',none,'|MovieLink|',none,'|ReleaseID|none,none,NULL|GenreID|non
e,none,NULL|RatingID|none,none,NULL|ScreenID|none,none,NULL|SoundID|none,none,NULL";

    // create the MM_fields and MM_columns arrays
    var MM_fields = MM_fieldsStr.split("|");
    var MM_columns = MM_columnsStr.split("|");

    // set the form values
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        MM_fields[i+1] = String(Request.Form(MM_fields[i]));
    }

    // append the query string to the redirect URL
    if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
        MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)? "?" : "&") + Request.QueryString;
```

```

    }
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
        var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
        if (formVal == "" || formVal == "undefined") {
            formVal = emptyVal;
        } else {
            if (altVal != "") {
                formVal = altVal;
            } else if (delim == "") { // escape quotes
                formVal = "" + formVal.replace(/'/g, "'") + "";
            } else {
                formVal = delim + formVal + delim;
            }
        }
        MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
        MM_dbValues += ((i != 0) ? "," : "") + formVal;
    }
    MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
    MM_dbValues + ")";

    if (!MM_abortEdit) {
        // execute the insert
        var MM_editCmd = Server.CreateObject('ADODB.Command');
        MM_editCmd.ActiveConnection = MM_editConnection;
        MM_editCmd.CommandText = MM_editQuery;
        MM_editCmd.Execute();
        MM_editCmd.ActiveConnection.Close();

        if (MM_editRedirectUrl) {
            Response.Redirect(MM_editRedirectUrl);
        }
    }

}
%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;

```

```

MovieSET.Source = "SELECT * FROM dbo.Movie ORDER BY MovieName ASC";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<%

var ReleaseSET = Server.CreateObject("ADODB.Recordset");
ReleaseSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ReleaseSET.Source = "SELECT * FROM dbo.Release ORDER BY ReleaseYear ASC";
ReleaseSET.CursorType = 0;
ReleaseSET.CursorLocation = 2;
ReleaseSET.LockType = 3;
ReleaseSET.Open();
var ReleaseSET_numRows = 0;
%>
<%

var GenreSET = Server.CreateObject("ADODB.Recordset");
GenreSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
GenreSET.Source = "SELECT * FROM dbo.Genre ORDER BY GenreName ASC";
GenreSET.CursorType = 0;
GenreSET.CursorLocation = 2;
GenreSET.LockType = 3;
GenreSET.Open();
var GenreSET_numRows = 0;
%>
<%

var RatingSET = Server.CreateObject("ADODB.Recordset");
RatingSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
RatingSET.Source = "SELECT * FROM dbo.Rating ORDER BY RatingID ASC";
RatingSET.CursorType = 0;
RatingSET.CursorLocation = 2;
RatingSET.LockType = 3;
RatingSET.Open();
var RatingSET_numRows = 0;
%>
<%

var ScreenSET = Server.CreateObject("ADODB.Recordset");
ScreenSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ScreenSET.Source = "SELECT * FROM dbo.Screen ORDER BY ScreenID ASC";
ScreenSET.CursorType = 0;
ScreenSET.CursorLocation = 2;
ScreenSET.LockType = 3;
ScreenSET.Open();
var ScreenSET_numRows = 0;
%>
<%

var SoundSET = Server.CreateObject("ADODB.Recordset");
SoundSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
SoundSET.Source = "SELECT * FROM dbo.Sound ORDER BY SoundID ASC";

```

```

SoundSET.CursorType = 0;
SoundSET.CursorLocation = 2;
SoundSET.LockType = 3;
SoundSET.Open();
var SoundSET_numRows = 0;
%>
<html>
<head>
<title>Add Movie (1 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<font color="#FFFFFF" size="5" face="Arial, Helvetica, sans-serif">Please Enter
The Movie Info (Page 1 of 4) </font>
<form name="AddMovieForm" method="POST" action="<%=MM_editAction%>">
  <table border="0" bgcolor="#000000">
    <tr>
      <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Movie
        Name</font></td>
      <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <input type="text" name="MovieNameText">
      </font></td>
    </tr>
    <tr>
      <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Movie
        Image</font></td>
      <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <input type="text" name="MovieImageText">
      </font></td>
    </tr>
    <tr>
      <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Movie
        Link</font></td>
      <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <input type="text" name="MovieLinkText">
      </font></td>
    </tr>
    <tr>
      <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Release
        Year</font></td>
      <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <select name="ReleaseSelect">
          <option value="NULL">Select Year</option>
        </select>
      </td>
    </tr>
  </table>
  while (!ReleaseSET.EOF)
  {
    <option value="<%= (ReleaseSET.Fields.Item("ReleaseID").Value)%>"
    ><%= (ReleaseSET.Fields.Item("ReleaseYear").Value)%></option>
    <%

```

```

ReleaseSET.MoveNext();
}
if (ReleaseSET.CursorType > 0) {
    if (!ReleaseSET.BOF) ReleaseSET.MoveFirst();
} else {
    ReleaseSET.Requery();
}
}%>

</select>
</font></td>
</tr>
<tr>
<td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Genre</font></td>
<td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
<select name="GenreSelect">
<option value="NULL">Select Genre</option>
<%
while (!GenreSET.EOF)
{
}%>
<option value="<%= (GenreSET.Fields.Item("GenreID").Value)%>"
><%= (GenreSET.Fields.Item("GenreName").Value)%></option>
<%
GenreSET.MoveNext();
}
if (GenreSET.CursorType > 0) {
    if (!GenreSET.BOF) GenreSET.MoveFirst();
} else {
    GenreSET.Requery();
}
}%>

</select>
</font></td>
</tr>
<tr>
<td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Rating</font></td>
<td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
<select name="RatingSelect">
<option value="NULL">Select Rating</option>
<%
while (!RatingSET.EOF)
{
}%>
<option value="<%= (RatingSET.Fields.Item("RatingID").Value)%>"
><%= (RatingSET.Fields.Item("RatingName").Value)%></option>
<%
RatingSET.MoveNext();
}
if (RatingSET.CursorType > 0) {

```

```

    if (!RatingSET.BOF) RatingSET.MoveFirst();
} else {
    RatingSET.Requery();
}
%>
    </select>
    </font></td>
</tr>
<tr>
    <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Screen
    Format</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <select name="ScreenSelect">
            <option value="NULL">Select Screen Format</option>
        <%
while (!ScreenSET.EOF)
{
%>
            <option value="<%= (ScreenSET.Fields.Item("ScreenID").Value)%>"
><%= (ScreenSET.Fields.Item("ScreenName").Value)%></option>
            <%
        ScreenSET.MoveNext();
    }
    if (ScreenSET.CursorType > 0) {
        if (!ScreenSET.BOF) ScreenSET.MoveFirst();
    } else {
        ScreenSET.Requery();
    }
%>
    </select>
    </font></td>
</tr>
<tr>
    <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Sound
    Format</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <select name="SoundSelect">
            <option value="NULL">Select Sound Format</option>
        <%
while (!SoundSET.EOF)
{
%>
            <option value="<%= (SoundSET.Fields.Item("SoundID").Value)%>"
><%= (SoundSET.Fields.Item("SoundName").Value)%></option>
            <%
        SoundSET.MoveNext();
    }
    if (SoundSET.CursorType > 0) {
        if (!SoundSET.BOF) SoundSET.MoveFirst();
    } else {
        SoundSET.Requery();
    }

```



```

// set the form action variable
var MM_editAction = Request.ServerVariables("URL");
if (Request.QueryString) {
    MM_editAction += "?" + Request.QueryString;
}

// boolean to abort record edit
var MM_abortEdit = false;

// query string to execute
var MM_editQuery = "";
%>
<%
// *** Insert Record: set variables

if (String(Request("MM_insert")) != "undefined") {

    var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
    var MM_editTable = "dbo.Formats_Movie";
    var MM_editRedirectUrl = "AddMovie3.asp";
    var MM_fieldsStr = "MovieFormatSelect|value|hiddenField1|value";
    var MM_columnsStr = "FormatsID|none,none,NULL|FMovieID|none,none,NULL";

    // create the MM_fields and MM_columns arrays
    var MM_fields = MM_fieldsStr.split("|");
    var MM_columns = MM_columnsStr.split("|");

    // set the form values
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        MM_fields[i+1] = String(Request.Form(MM_fields[i]));
    }

    // append the query string to the redirect URL
    if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
        MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)?"?": "&") + Request.QueryString;
    }
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
    }
}

```

```

var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
if (formVal == "" || formVal == "undefined") {
    formVal = emptyVal;
} else {
    if (altVal != "") {
        formVal = altVal;
    } else if (delim == "") { // escape quotes
        formVal = "" + formVal.replace(/'/g, "'") + "";
    } else {
        formVal = delim + formVal + delim;
    }
}
MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
MM_dbValues += ((i != 0) ? "," : "") + formVal;
}
MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
MM_dbValues + ")";

if (!MM_abortEdit) {
    // execute the insert
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
    MM_editCmd.ActiveConnection.Close();

    if (MM_editRedirectUrl) {
        Response.Redirect(MM_editRedirectUrl);
    }
}

%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieSET.Source = "SELECT MAX(MovieID) AS MaxMovieID FROM dbo.Movie";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<%
var MovieFormatsSET = Server.CreateObject("ADODB.Recordset");
MovieFormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieFormatsSET.Source = "SELECT * FROM dbo.Formats ORDER BY FormatsID ASC";
MovieFormatsSET.CursorType = 0;
MovieFormatsSET.CursorLocation = 2;
MovieFormatsSET.LockType = 3;
MovieFormatsSET.Open();

```

```

var MovieFormatsSET_numRows = 0;
%>
<%
var ActorSET = Server.CreateObject("ADODB.Recordset");
ActorSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ActorSET.Source = "SELECT ActorID, ActorFName+' '+ActorLName AS ActorFullName FROM
dbo.Actor ORDER BY ActorFullName ASC";
ActorSET.CursorType = 0;
ActorSET.CursorLocation = 2;
ActorSET.LockType = 3;
ActorSET.Open();
var ActorSET_numRows = 0;
%>
<html>
<head>
<title>Add Movie (2 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#000000">
<center>
<font color="#FFFFFF" size="5" face="Arial, Helvetica, sans-serif"> Please Enter
The Movie Format (Page 2 of 4) </font>
<form name="form1" method="POST" action="<%=MM_editAction%>">
<table border="0">
<tr>
<td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Movie
Format</font></td>
<td> <font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
<select name="MovieFormatSelect">
<option value="NULL">Select Movie Format</option>
<%
while (!MovieFormatsSET.EOF) {
%>
<option
value="<%= (MovieFormatsSET.Fields.Item("FormatsID").Value)%>"><%= (MovieFormatsSET.Fields.
Item("FormatsName").Value)%></option>
<%
MovieFormatsSET.MoveNext();
}
if (MovieFormatsSET.CursorType > 0) {
if (!MovieFormatsSET.BOF) MovieFormatsSET.MoveFirst();
} else {
MovieFormatsSET.Requery();
}
%>
</select>
<input type="hidden" name="hiddenField1"
value="<%= (MovieSET.Fields.Item("MaxMovieID").Value)%>">
</font></td>
</tr>
<tr>

```



```

var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
var MM_editTable = "dbo.Location_Movie";
var MM_editRedirectUrl = "AddMovie4.asp";
var MM_fieldsStr = "MovieLocationSelect|value|hiddenField2|value";
var MM_columnsStr = "LocationID|none,none,NULL|LMovieID|none,none,NULL";

// create the MM_fields and MM_columns arrays
var MM_fields = MM_fieldsStr.split("|");
var MM_columns = MM_columnsStr.split("|");

// set the form values
for (var i=0; i+1 < MM_fields.length; i+=2) {
    MM_fields[i+1] = String(Request.Form(MM_fields[i]));
}

// append the query string to the redirect URL
if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
    MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf('?') == -1)? "?" : "&") + Request.QueryString;
}
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
        var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
        if (formVal == "" || formVal == "undefined") {
            formVal = emptyVal;
        } else {
            if (altVal != "") {
                formVal = altVal;
            } else if (delim == "") { // escape quotes
                formVal = "" + formVal.replace(/'/g, "'") + "";
            } else {
                formVal = delim + formVal + delim;
            }
        }
        MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
        MM_dbValues += ((i != 0) ? "," : "") + formVal;
    }
    MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
    MM_dbValues + ")";
}

```

```

if (!MM_abortEdit) {
    // execute the insert
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
    MM_editCmd.ActiveConnection.Close();

    if (MM_editRedirectUrl) {
        Response.Redirect(MM_editRedirectUrl);
    }
}

}
%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieSET.Source = "SELECT MAX(MovieID) AS MaxMovieID FROM dbo.Movie";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<%
var ActorSET = Server.CreateObject("ADODB.Recordset");
ActorSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ActorSET.Source = "SELECT ActorID, ActorFName+' '+ActorLName AS ActorFullName FROM
dbo.Actor ORDER BY ActorFullName ASC";
ActorSET.CursorType = 0;
ActorSET.CursorLocation = 2;
ActorSET.LockType = 3;
ActorSET.Open();
var ActorSET_numRows = 0;
%>
<%
var MovieLocation = Server.CreateObject("ADODB.Recordset");
MovieLocation.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieLocation.Source = "SELECT * FROM dbo.Location ORDER BY LocationID ASC";
MovieLocation.CursorType = 0;
MovieLocation.CursorLocation = 2;
MovieLocation.LockType = 3;
MovieLocation.Open();
var MovieLocation_numRows = 0;
%>
<html>
<head>
<title>Add Movie (3 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

```



```

MovieSET.Close();
%>
<%
ActorSET.Close();
%>
<%
MovieLocation.Close();
%>

```

7.4 Add Movie (4 of 4) Part A (AddMovie4.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
// *** Edit Operations: declare variables

// set the form action variable
var MM_editAction = Request.ServerVariables("URL");
if (Request.QueryString) {
    MM_editAction += "?" + Request.QueryString;
}

// boolean to abort record edit
var MM_abortEdit = false;

// query string to execute
var MM_editQuery = "";
%>
<%
// *** Insert Record: set variables

if (String(Request("MM_insert")) != "undefined") {

    var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
    var MM_editTable = "dbo.Actor_Movie";
    var MM_editRedirectUrl = "AddMovie5.asp";
    var MM_fieldsStr = "Actor1Select|value|hiddenField3|value";
    var MM_columnsStr = "ActorID|none,none,NULL|AMovieID|none,none,NULL";

    // create the MM_fields and MM_columns arrays
    var MM_fields = MM_fieldsStr.split("|");
    var MM_columns = MM_columnsStr.split("|");

    // set the form values
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        MM_fields[i+1] = String(Request.Form(MM_fields[i]));
    }

    // append the query string to the redirect URL
    if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
        MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf('?') == -1)? "?" : "&") + Request.QueryString;
    }
}

```

```

    }
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
        var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
        if (formVal == "" || formVal == "undefined") {
            formVal = emptyVal;
        } else {
            if (altVal != "") {
                formVal = altVal;
            } else if (delim == "") { // escape quotes
                formVal = "" + formVal.replace(/'/g, "'") + "";
            } else {
                formVal = delim + formVal + delim;
            }
        }
        MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
        MM_dbValues += ((i != 0) ? "," : "") + formVal;
    }
    MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
    MM_dbValues + ")";

    if (!MM_abortEdit) {
        // execute the insert
        var MM_editCmd = Server.CreateObject('ADODB.Command');
        MM_editCmd.ActiveConnection = MM_editConnection;
        MM_editCmd.CommandText = MM_editQuery;
        MM_editCmd.Execute();
        MM_editCmd.ActiveConnection.Close();

        if (MM_editRedirectUrl) {
            Response.Redirect(MM_editRedirectUrl);
        }
    }

}
%>
<%
var ActorSET = Server.CreateObject("ADODB.Recordset");
ActorSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;

```

```

ActorSET.Source = "SELECT ActorID, ActorFName+' '+ActorLName AS ActorFullName FROM
dbo.Actor ORDER BY ActorFullName ASC";
ActorSET.CursorType = 0;
ActorSET.CursorLocation = 2;
ActorSET.LockType = 3;
ActorSET.Open();
var ActorSET_numRows = 0;
%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieSET.Source = "SELECT MAX(MovieID) AS MaxMovieID FROM dbo.Movie";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<html>
<head>
<title>Add Movie (4 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<font color="#FFFFFF" size="5" face="Arial, Helvetica, sans-serif">Please Enter
The Actor(s) (Page 4 of 4) </font>
<form name="form1" method="POST" action="<%=MM_editAction%>">
  <table width="360" border="0">
    <tr>
      <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Add
        Actors</font></td>
      <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <select name="Actor1Select">
          <option value="NULL">Select Actor</option>
          <%
while (!ActorSET.EOF)
{
%>
          <option value="<%= (ActorSET.Fields.Item("ActorID").Value)%>"
><%= (ActorSET.Fields.Item("ActorFullName").Value)%></option>
          <%
        ActorSET.MoveNext();
      }
if (ActorSET.CursorType > 0) {
  if (!ActorSET.BOF) ActorSET.MoveFirst();
} else {
  ActorSET.Requery();
}
%>
        </select>

```

```

        <input type="hidden" name="hiddenField3"
value="<%=(MovieSET.Fields.Item("MaxMovieID").Value)%>">
        <input type="submit" name="Submit" value="Add Actor">
        </font></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td colspan="2"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><a
href="AddActor.asp" target="_new">Add
        New Actor</a></font></td>
    </tr>
</table>
<font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
<input type="hidden" name="MM_insert" value="true">
</font>
</form>
</center>
</body>
</html>
<%
ActorSET.Close();
%>
<%
MovieSET.Close();
%>

```

7.5 Add Movie (4 of 4) Part B (AddMovie5.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
// *** Edit Operations: declare variables

// set the form action variable
var MM_editAction = Request.ServerVariables("URL");
if (Request.QueryString) {
    MM_editAction += "?" + Request.QueryString;
}

// boolean to abort record edit
var MM_abortEdit = false;

// query string to execute
var MM_editQuery = "";
%>
<%
// *** Insert Record: set variables

```

```

if (String(Request("MM_insert")) != "undefined") {

    var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
    var MM_editTable = "dbo.Actor_Movie";
    var MM_editRedirectUrl = "";
    var MM_fieldsStr = "Actor1Select|value|hiddenField3|value";
    var MM_columnsStr = "ActorID|none,none,NULL|AMovieID|none,none,NULL";

    // create the MM_fields and MM_columns arrays
    var MM_fields = MM_fieldsStr.split("|");
    var MM_columns = MM_columnsStr.split("|");

    // set the form values
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        MM_fields[i+1] = String(Request.Form(MM_fields[i]));
    }

    // append the query string to the redirect URL
    if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
        MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf('?') == -1)? "?" : "&") + Request.QueryString;
    }
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
        var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
        if (formVal == "" || formVal == "undefined") {
            formVal = emptyVal;
        } else {
            if (altVal != "") {
                formVal = altVal;
            } else if (delim == "") { // escape quotes
                formVal = "" + formVal.replace(/'/g, "'") + "";
            } else {
                formVal = delim + formVal + delim;
            }
        }
    }
    MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
    MM_dbValues += ((i != 0) ? "," : "") + formVal;
}

```

```

MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
MM_dbValues + ")";

if (!MM_abortEdit) {
    // execute the insert
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
    MM_editCmd.ActiveConnection.Close();

    if (MM_editRedirectUrl) {
        Response.Redirect(MM_editRedirectUrl);
    }
}

}
%>
<%
var ActorSET = Server.CreateObject("ADODB.Recordset");
ActorSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ActorSET.Source = "SELECT ActorID, ActorFName+' '+ActorLName AS ActorFullName FROM
dbo.Actor ORDER BY ActorFullName ASC";
ActorSET.CursorType = 0;
ActorSET.CursorLocation = 2;
ActorSET.LockType = 3;
ActorSET.Open();
var ActorSET_numRows = 0;
%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieSET.Source = "SELECT MAX(MovieID) AS MaxMovieID FROM dbo.Movie";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<html>
<head>
<title>Add Movie (4 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<font color="#FFFFFF" size="5" face="Arial, Helvetica, sans-serif">Please Enter
The Actor(s) (Page 4 of 4) </font>
<form name="form1" method="POST" action="<%=MM_editAction%>">
    <table width="360" border="0">
        <tr>

```

```

        <td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Add
        Actors</font></td>
        <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
        <select name="Actor1Select">
            <option value="NULL">Select Another Actor?</option>
            <%
while (!ActorSET.EOF)
{
%>
            <option value="<%= (ActorSET.Fields.Item("ActorID").Value)%>"
><%= (ActorSET.Fields.Item("ActorFullName").Value)%></option>
            <%
            ActorSET.MoveNext();
}
if (ActorSET.CursorType > 0) {
    if (!ActorSET.BOF) ActorSET.MoveFirst();
} else {
    ActorSET.Requery();
}
%>
        </select>
        <input type="hidden" name="hiddenField3"
value="<%= (MovieSET.Fields.Item("MaxMovieID").Value)%>">
        <input type="submit" name="Submit" value="Add Actor">
        </font></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td colspan="2"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><a
href="AddActor.asp" target="_new">Add
        New Actor</a></font></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><a href="ViewMovie.asp"><font
size="+2">Finish
        Adding Movie</font></a></font></td>
    </tr>
</table>
<font color="#FFFFFF" face="Arial, Helvetica, sans-serif">
<input type="hidden" name="MM_insert" value="true">
</font>
</form>
</center>

```

```

</body>
</html>
<%
ActorSET.Close();
%>
<%
MovieSET.Close();
%>

```

7.6 Add an Actor (AddActor.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
// *** Edit Operations: declare variables

// set the form action variable
var MM_editAction = Request.ServerVariables("URL");
if (Request.QueryString) {
    MM_editAction += "?" + Request.QueryString;
}

// boolean to abort record edit
var MM_abortEdit = false;

// query string to execute
var MM_editQuery = "";
%>
<%
// *** Insert Record: set variables

if (String(Request("MM_insert")) != "undefined") {

    var MM_editConnection = MM_MovieCollectionDNSasp_STRING;
    var MM_editTable = "dbo.Actor";
    var MM_editRedirectUrl = "AddActor.asp";
    var MM_fieldsStr =
"FirstNameText|value|LastNameText|value|ImageNameText|value|ActorLinkText|value";
    var MM_columnsStr =
"ActorFName|',none,'|ActorLName|',none,'|ActorImage|',none,'|ActorLink|',none,'";

    // create the MM_fields and MM_columns arrays
    var MM_fields = MM_fieldsStr.split("|");
    var MM_columns = MM_columnsStr.split("|");

    // set the form values
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        MM_fields[i+1] = String(Request.Form(MM_fields[i]));
    }

    // append the query string to the redirect URL

```

```

if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
    MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)? "?" : "&") + Request.QueryString;
}
}
%>
<%
// *** Insert Record: construct a sql insert statement and execute it

if (String(Request("MM_insert")) != "undefined") {

    // create the sql insert statement
    var MM_tableValues = "", MM_dbValues = "";
    for (var i=0; i+1 < MM_fields.length; i+=2) {
        var formVal = MM_fields[i+1];
        var MM_typesArray = MM_columns[i+1].split(",");
        var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
        var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
        var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
        if (formVal == "" || formVal == "undefined") {
            formVal = emptyVal;
        } else {
            if (altVal != "") {
                formVal = altVal;
            } else if (delim == "") { // escape quotes
                formVal = "" + formVal.replace(/'/g, "'") + "";
            } else {
                formVal = delim + formVal + delim;
            }
        }
        MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
        MM_dbValues += ((i != 0) ? "," : "") + formVal;
    }
    MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values (" +
    MM_dbValues + ")";

    if (!MM_abortEdit) {
        // execute the insert
        var MM_editCmd = Server.CreateObject('ADODB.Command');
        MM_editCmd.ActiveConnection = MM_editConnection;
        MM_editCmd.CommandText = MM_editQuery;
        MM_editCmd.Execute();
        MM_editCmd.ActiveConnection.Close();

        if (MM_editRedirectUrl) {
            Response.Redirect(MM_editRedirectUrl);
        }
    }
}
%>
<%

```

```

var ActorSET = Server.CreateObject("ADODB.Recordset");
ActorSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ActorSET.Source = "SELECT * FROM dbo.Actor";
ActorSET.CursorType = 0;
ActorSET.CursorLocation = 2;
ActorSET.LockType = 3;
ActorSET.Open();
var ActorSET_numRows = 0;
%>
<html>
<head>
<title>Add Actor(s)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<font color="#FFFFFF" face="Arial, Helvetica, sans-serif"> <font size="5">Please
Add The Actor(s)</font><br>
<form name="form1" method="POST" action="<%=MM_editAction%>">
<table border="0">
<tr>
<td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">First
Name</font></td>
<td><input type="text" name="FirstNameText"></td>
</tr>
<tr>
<td bgcolor="#999999"><font color="#FFFFFF" face="Arial, Helvetica, sans-serif">Last
Name</font></td>
<td><input type="text" name="LastNameText"></td>
</tr>
<tr>
<td bgcolor="#999999">Image Name</td>
<td><input type="text" name="ImageNameText"></td>
</tr>
<tr>
<td bgcolor="#999999">Actor Link</td>
<td><input type="text" name="ActorLinkText"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr align="right">
<td colspan="2"><input type="reset" name="Reset" value="Clear Form">
<input type="submit" name="Submit2" value="Add Actor"></td>
</tr>
</table>
<input type="hidden" name="MM_insert" value="true">
</form>
<a href="index.htm"><font size="5">Finished Adding Actors</font></a></font>
</center>

```

```

</body>
</html>
<%
ActorSET.Close();
%>

```

7.7 Edit a Movie (1 of 4) (EditMovie.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
var ViewMovieSET = Server.CreateObject("ADODB.Recordset");
ViewMovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ViewMovieSET.Source = "SELECT M.MovieID, M.MovieName+' '+F.FormatsName AS
MovieAndFormat FROM dbo.Movie AS M, dbo.Formats_Movie AS FM, dbo.Formats AS F WHERE
M.MovieID=FM.FMovieID AND FM.FormatsID=F.FormatsID ORDER BY MovieName ASC";
ViewMovieSET.CursorType = 0;
ViewMovieSET.CursorLocation = 2;
ViewMovieSET.LockType = 3;
ViewMovieSET.Open();
var ViewMovieSET_numRows = 0;
%>
<%
// *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

// set the record count
var ViewMovieSET_total = ViewMovieSET.RecordCount;

// set the number of rows displayed on this page
if (ViewMovieSET_numRows < 0) { // if repeat region set to all records
    ViewMovieSET_numRows = ViewMovieSET_total;
} else if (ViewMovieSET_numRows == 0) { // if no repeat regions
    ViewMovieSET_numRows = 1;
}

// set the first and last displayed record
var ViewMovieSET_first = 1;
var ViewMovieSET_last = ViewMovieSET_first + ViewMovieSET_numRows - 1;

// if we have the correct record count, check the other stats
if (ViewMovieSET_total != -1) {
    ViewMovieSET_numRows = Math.min(ViewMovieSET_numRows, ViewMovieSET_total);
    ViewMovieSET_first = Math.min(ViewMovieSET_first, ViewMovieSET_total);
    ViewMovieSET_last = Math.min(ViewMovieSET_last, ViewMovieSET_total);
}
%>
<%
// *** Recordset Stats: if we don't know the record count, manually count them

if (ViewMovieSET_total == -1) {

```

```

// count the total records by iterating through the recordset
for (ViewMovieSET_total=0; !ViewMovieSET.EOF; ViewMovieSET.MoveNext()) {
    ViewMovieSET_total++;
}

// reset the cursor to the beginning
if (ViewMovieSET.CursorType > 0) {
    if (!ViewMovieSET.BOF) ViewMovieSET.MoveFirst();
} else {
    ViewMovieSET.Requery();
}

// set the number of rows displayed on this page
if (ViewMovieSET_numRows < 0 || ViewMovieSET_numRows > ViewMovieSET_total) {
    ViewMovieSET_numRows = ViewMovieSET_total;
}

// set the first and last displayed record
ViewMovieSET_last = Math.min(ViewMovieSET_first + ViewMovieSET_numRows - 1,
ViewMovieSET_total);
ViewMovieSET_first = Math.min(ViewMovieSET_first, ViewMovieSET_total);
}
%>
<html>
<head>
<title>Edit Movie</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#000000">
<center>
<form name="form1" method="<%= (ViewMovieSET.Fields.Item("MovieID").Value) %>"
action="EditMovie2.asp">
    <font color="#FFFFFF" face="Arial, Helvetica, sans-serif"> <font size="6">Select
    A Movie to Edit</font><br>
    <br>
    <select name="MovieID">
        <option value="NULL">Select Movie</option>
        <%
while (!ViewMovieSET.EOF)
{
%>
    <option
value="<%= (ViewMovieSET.Fields.Item("MovieID").Value) %>"><%= (ViewMovieSET.Fields.Item("
MovieAndFormat").Value) %></option>
    <%
    ViewMovieSET.MoveNext();
}
if (ViewMovieSET.CursorType > 0) {
    if (!ViewMovieSET.BOF) ViewMovieSET.MoveFirst();
} else {
    ViewMovieSET.Requery();

```



```

// create the MM_fields and MM_columns arrays
var MM_fields = MM_fieldsStr.split("|");
var MM_columns = MM_columnsStr.split("|");

// set the form values
for (var i=0; i+1 < MM_fields.length; i+=2) {
    MM_fields[i+1] = String(Request.Form(MM_fields[i]));
}

// append the query string to the redirect URL
if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.length > 0) {
    MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf('?') == -1)?'?' : '&') + Request.QueryString;
}
}
%>
<%
// *** Update Record: construct a sql update statement and execute it

if (String(Request("MM_update")) != "undefined" &&
    String(Request("MM_recordId")) != "undefined") {

// create the sql update statement
MM_editQuery = "update " + MM_editTable + " set ";
for (var i=0; i+1 < MM_fields.length; i+=2) {
    var formVal = MM_fields[i+1];
    var MM_typesArray = MM_columns[i+1].split(",");
    var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
    var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
    var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
    if (formVal == "" || formVal == "undefined") {
        formVal = emptyVal;
    } else {
        if (altVal != "") {
            formVal = altVal;
        } else if (delim == "") { // escape quotes
            formVal = "" + formVal.replace(/'/g, "'") + "";
        } else {
            formVal = delim + formVal + delim;
        }
    }
}
MM_editQuery += ((i != 0) ? ", " : "") + MM_columns[i] + " = " + formVal;
}
MM_editQuery += " where " + MM_editColumn + " = " + MM_recordId;

if (!MM_abortEdit) {
    // execute the update
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
}

```

```

MM_editCmd.ActiveConnection.Close();

if (MM_editRedirectUrl) {
    Response.Redirect(MM_editRedirectUrl);
}
}

}
%>
<%
var MovieSET__MMColParam = "1";
if(String(Request.QueryString("MovieID")) != "undefined") {
    MovieSET__MMColParam = String(Request.QueryString("MovieID"));
}
%>
<%
var MovieSET = Server.CreateObject("ADODB.Recordset");
MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieSET.Source = "SELECT * FROM dbo.Movie WHERE MovieID = "+
MovieSET__MMColParam.replace(/'/g, "''") + "'";
MovieSET.CursorType = 0;
MovieSET.CursorLocation = 2;
MovieSET.LockType = 3;
MovieSET.Open();
var MovieSET_numRows = 0;
%>
<%
var ReleaseSET = Server.CreateObject("ADODB.Recordset");
ReleaseSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ReleaseSET.Source = "SELECT * FROM dbo.Release ORDER BY ReleaseYear ASC";
ReleaseSET.CursorType = 0;
ReleaseSET.CursorLocation = 2;
ReleaseSET.LockType = 3;
ReleaseSET.Open();
var ReleaseSET_numRows = 0;
%>
<%
var GenreSET = Server.CreateObject("ADODB.Recordset");
GenreSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
GenreSET.Source = "SELECT * FROM dbo.Genre ORDER BY GenreName ASC";
GenreSET.CursorType = 0;
GenreSET.CursorLocation = 2;
GenreSET.LockType = 3;
GenreSET.Open();
var GenreSET_numRows = 0;
%>
<%
var RatingSET = Server.CreateObject("ADODB.Recordset");
RatingSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
RatingSET.Source = "SELECT * FROM dbo.Rating ORDER BY RatingID ASC";
RatingSET.CursorType = 0;

```

```

RatingSET.CursorLocation = 2;
RatingSET.LockType = 3;
RatingSET.Open();
var RatingSET_numRows = 0;
%>
<%
var ScreenSET = Server.CreateObject("ADODB.Recordset");
ScreenSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ScreenSET.Source = "SELECT * FROM dbo.Screen ORDER BY ScreenID ASC";
ScreenSET.CursorType = 0;
ScreenSET.CursorLocation = 2;
ScreenSET.LockType = 3;
ScreenSET.Open();
var ScreenSET_numRows = 0;
%>
<%
var SoundID = Server.CreateObject("ADODB.Recordset");
SoundID.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
SoundID.Source = "SELECT * FROM dbo.Sound ORDER BY SoundID ASC";
SoundID.CursorType = 0;
SoundID.CursorLocation = 2;
SoundID.LockType = 3;
SoundID.Open();
var SoundID_numRows = 0;
%>
<html>
<head>
<title>Edit Movie (1 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<font color="#FFFFFF" size="6" face="Arial, Helvetica, sans-serif"> Edit Movie
(Page 1 of 4)</font>
<form method="POST" action="<%=MM_editAction%>" name="form1">
<table align="center">
<tr valign="baseline">
<td nowrap align="right">MovieName:</td>
<td> <input type="text" name="MovieName"
value="<%= (MovieSET.Fields.Item("MovieName").Value)%>" size="32">
</td>
</tr>
<tr valign="baseline">
<td nowrap align="right">MovieImage:</td>
<td> <input type="text" name="MovieImage"
value="<%= (MovieSET.Fields.Item("MovieImage").Value)%>" size="32">
</td>
</tr>
<tr valign="baseline">
<td nowrap align="right">MovieLink:</td>

```

```

        <td> <input type="text" name="MovieLink"
value="<%= (MovieSET.Fields.Item("MovieLink").Value)%>" size="32">
        </td>
    </tr>
    <tr valign="baseline">
        <td nowrap align="right">ReleaseID:</td>
        <td> <select name="ReleaseID">
            <%=
while (!ReleaseSET.EOF) {
%>
            <option value="<%= (ReleaseSET.Fields.Item("ReleaseID").Value)%>"
<%= ((ReleaseSET.Fields.Item("ReleaseID").Value ==
MovieSET.Fields.Item("ReleaseID").Value)? "SELECTED": "")%>
><%= (ReleaseSET.Fields.Item("ReleaseYear").Value)%></option>
            <%=
ReleaseSET.MoveNext();
}
if (ReleaseSET.CursorType > 0) {
    if (!ReleaseSET.BOF) ReleaseSET.MoveFirst();
} else {
    ReleaseSET.Requery();
}
%>
        </select> </td>
    </tr>
    <tr valign="baseline">
        <td nowrap align="right">GenreID:</td>
        <td> <select name="GenreID">
            <%=
while (!GenreSET.EOF) {
%>
            <option value="<%= (GenreSET.Fields.Item("GenreID").Value)%>"
<%= ((GenreSET.Fields.Item("GenreID").Value ==
MovieSET.Fields.Item("GenreID").Value)? "SELECTED": "")%>
><%= (GenreSET.Fields.Item("GenreName").Value)%></option>
            <%=
GenreSET.MoveNext();
}
if (GenreSET.CursorType > 0) {
    if (!GenreSET.BOF) GenreSET.MoveFirst();
} else {
    GenreSET.Requery();
}
%>
        </select> </td>
    </tr>
    <tr valign="baseline">
        <td nowrap align="right">RatingID:</td>
        <td> <select name="RatingID">
            <%=
while (!RatingSET.EOF) {

```

```

%>
    <option value="<%= (RatingSET.Fields.Item("RatingID").Value)%>"
<%= ((RatingSET.Fields.Item("RatingID").Value ==
MovieSET.Fields.Item("RatingID").Value)? "SELECTED": "")%>
><%= (RatingSET.Fields.Item("RatingName").Value)%></option>
    <%
        RatingSET.MoveNext();
    }
    if (RatingSET.CursorType > 0) {
        if (!RatingSET.BOF) RatingSET.MoveFirst();
    } else {
        RatingSET.Requery();
    }
%>
    </select> </td>
</tr>
<tr valign="baseline">
    <td nowrap align="right">ScreenID:</td>
    <td> <select name="ScreenID">
        <%
while (!ScreenSET.EOF) {
%>
    <option value="<%= (ScreenSET.Fields.Item("ScreenID").Value)%>"
<%= ((ScreenSET.Fields.Item("ScreenID").Value ==
MovieSET.Fields.Item("ScreenID").Value)? "SELECTED": "")%>
><%= (ScreenSET.Fields.Item("ScreenName").Value)%></option>
    <%
        ScreenSET.MoveNext();
    }
    if (ScreenSET.CursorType > 0) {
        if (!ScreenSET.BOF) ScreenSET.MoveFirst();
    } else {
        ScreenSET.Requery();
    }
%>
    </select> </td>
</tr>
<tr valign="baseline">
    <td nowrap align="right">SoundID:</td>
    <td> <select name="SoundID">
        <%
while (!SoundID.EOF) {
%>
    <option value="<%= (SoundID.Fields.Item("SoundID").Value)%>"
<%= ((SoundID.Fields.Item("SoundID").Value ==
MovieSET.Fields.Item("SoundID").Value)? "SELECTED": "")%>
><%= (SoundID.Fields.Item("SoundName").Value)%></option>
    <%
        SoundID.MoveNext();
    }
    if (SoundID.CursorType > 0) {

```

```

    if (!SoundID.BOF) SoundID.MoveFirst();
} else {
    SoundID.Requery();
}
%>
    </select> </td>
</tr>
<tr valign="baseline">
    <td nowrap align="right">&nbsp;</td>
    <td> <input type="submit" value="Update Record"> </td>
</tr>
</table>
<input type="hidden" name="MM_update" value="true">
<input type="hidden" name="MM_recordId" value="<%= MovieSET.Fields.Item("MovieID").Value
%>">
</form>
<p>&nbsp;</p>
<p>&nbsp;</p>
</center>
</body>
</html>
<%
MovieSET.Close();
%>
<%
ReleaseSET.Close();
%>
<%
GenreSET.Close();
%>
<%
RatingSET.Close();
%>
<%
ScreenSET.Close();
%>
<%
SoundID.Close();
%>

```

7.9 Edit a Movie (3 of 4) (EditMovie3.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
var Formats_MovieSET__MMColParam = "1";
if(String(Request.QueryString("MovieID")) != "undefined") {
    Formats_MovieSET__MMColParam = String(Request.QueryString("MovieID"));
}
%>
<%

```

```

var Formats_MovieSET = Server.CreateObject("ADODB.Recordset");
Formats_MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
Formats_MovieSET.Source = "SELECT * FROM dbo.Formats_Movie WHERE FMovieID = "+
Formats_MovieSET__MMColParam.replace("/g, """) + """;
Formats_MovieSET.CursorType = 0;
Formats_MovieSET.CursorLocation = 2;
Formats_MovieSET.LockType = 3;
Formats_MovieSET.Open();
var Formats_MovieSET_numRows = 0;
%>
<%
var FormatsSET = Server.CreateObject("ADODB.Recordset");
FormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
FormatsSET.Source = "SELECT * FROM dbo.Formats ORDER BY FormatsID ASC";
FormatsSET.CursorType = 0;
FormatsSET.CursorLocation = 2;
FormatsSET.LockType = 3;
FormatsSET.Open();
var FormatsSET_numRows = 0;
%>
<%
var MovieFormatsSET__MMColParam = "0";
if(String(Request.QueryString("MovieID")) != "undefined") {
    MovieFormatsSET__MMColParam = String(Request.QueryString("MovieID"));
}
%>
<%
var MovieFormatsSET = Server.CreateObject("ADODB.Recordset");
MovieFormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieFormatsSET.Source = "SELECT M.MovieName+' '+F.FormatsName FROM dbo.Movie AS M,
dbo.Formats_Movie AS FM, dbo.Formats AS F WHERE MovieID = "+
MovieFormatsSET__MMColParam.replace("/g, """) + " AND M.MovieID=FM.FMovieID AND
FM.FormatsID=F.FormatsID";

MovieFormatsSET.CursorType = 0;
MovieFormatsSET.CursorLocation = 2;
MovieFormatsSET.LockType = 3;
MovieFormatsSET.Open();
var MovieFormatsSET_numRows = 0;
%>
<html>
<head>
<title>Edit Movie (2 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
    <font color="#FFFFFF" size="6" face="Arial, Helvetica, sans-serif"> Edit Movie
    (Page 2 of 4)</font> <br>
    <%= (MovieSET.Fields.Item("MovieName").Value) %>
</center>

```

```

</body>
</html>
<%
Formats_MovieSET.Close();
%>
<%
FormatsSET.Close();
%>
<%
MovieFormatsSET.Close();
%>

```

7.10 Edit a Movie (4 of 4) (EditMovie4.asp)

```

<%@LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
var Formats_MovieSET__MMColParam = "1";
if(String(Request.QueryString("MovieID")) != "undefined") {
    Formats_MovieSET__MMColParam = String(Request.QueryString("MovieID"));
}
%>
<%
var Formats_MovieSET = Server.CreateObject("ADODB.Recordset");
Formats_MovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
Formats_MovieSET.Source = "SELECT * FROM dbo.Formats_Movie WHERE FMovieID = "+
Formats_MovieSET__MMColParam.replace(/'/g, "'") + "'";
Formats_MovieSET.CursorType = 0;
Formats_MovieSET.CursorLocation = 2;
Formats_MovieSET.LockType = 3;
Formats_MovieSET.Open();
var Formats_MovieSET_numRows = 0;
%>
<%
var FormatsSET = Server.CreateObject("ADODB.Recordset");
FormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
FormatsSET.Source = "SELECT * FROM dbo.Formats ORDER BY FormatsID ASC";
FormatsSET.CursorType = 0;
FormatsSET.CursorLocation = 2;
FormatsSET.LockType = 3;
FormatsSET.Open();
var FormatsSET_numRows = 0;
%>
<%
var MovieFormatsSET__MMColParam = "0";
if(String(Request.QueryString("MovieID")) != "undefined") {
    MovieFormatsSET__MMColParam = String(Request.QueryString("MovieID"));
}
%>
<%
var MovieFormatsSET = Server.CreateObject("ADODB.Recordset");

```

```

MovieFormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
MovieFormatsSET.Source = "SELECT M.MovieName+' '+F.FormatsName FROM dbo.Movie AS M,
dbo.Formats_Movie AS FM, dbo.Formats AS F WHERE MovieID = "+
MovieFormatsSET__MMColParam.replace("/g, "") + " AND M.MovieID=FM.FMMovieID AND
FM.FormatsID=F.FormatsID";

MovieFormatsSET.CursorType = 0;
MovieFormatsSET.CursorLocation = 2;
MovieFormatsSET.LockType = 3;
MovieFormatsSET.Open();
var MovieFormatsSET_numRows = 0;
%>
<html>
<head>
<title>Edit Movie (2 of 4)</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
  <font color="#FFFFFF" size="6" face="Arial, Helvetica, sans-serif"> Edit Movie
  (Page 2 of 4)</font> <br>
  <%= (MovieSET.Fields.Item("MovieName").Value)%>
</center>
</body>
</html>
<%
Formats_MovieSET.Close();
%>
<%
FormatsSET.Close();
%>
<%
MovieFormatsSET.Close();
%>

```

7.11 View Movie (ViewMovie.asp)

```

<%= @LANGUAGE="JAVASCRIPT"%>
<!--#include file="../Connections/MovieCollectionDNSasp.asp" -->
<%
var ViewMovieSET = Server.CreateObject("ADODB.Recordset");
ViewMovieSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ViewMovieSET.Source = "SELECT M.MovieID, M.MovieName+' '+F.FormatsName AS
MovieAndFormat FROM dbo.Movie AS M, dbo.Formats_Movie AS FM, dbo.Formats AS F WHERE
M.MovieID=FM.FMMovieID AND FM.FormatsID=F.FormatsID ORDER BY MovieName ASC";
ViewMovieSET.CursorType = 0;
ViewMovieSET.CursorLocation = 2;
ViewMovieSET.LockType = 3;
ViewMovieSET.Open();
var ViewMovieSET_numRows = 0;
%>

```

```

<%
// *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

// set the record count
var ViewMovieSET_total = ViewMovieSET.RecordCount;

// set the number of rows displayed on this page
if (ViewMovieSET_numRows < 0) { // if repeat region set to all records
    ViewMovieSET_numRows = ViewMovieSET_total;
} else if (ViewMovieSET_numRows == 0) { // if no repeat regions
    ViewMovieSET_numRows = 1;
}

// set the first and last displayed record
var ViewMovieSET_first = 1;
var ViewMovieSET_last = ViewMovieSET_first + ViewMovieSET_numRows - 1;

// if we have the correct record count, check the other stats
if (ViewMovieSET_total != -1) {
    ViewMovieSET_numRows = Math.min(ViewMovieSET_numRows, ViewMovieSET_total);
    ViewMovieSET_first = Math.min(ViewMovieSET_first, ViewMovieSET_total);
    ViewMovieSET_last = Math.min(ViewMovieSET_last, ViewMovieSET_total);
}
%>
<%
// *** Recordset Stats: if we don't know the record count, manually count them

if (ViewMovieSET_total == -1) {

    // count the total records by iterating through the recordset
    for (ViewMovieSET_total=0; !ViewMovieSET.EOF; ViewMovieSET.MoveNext()) {
        ViewMovieSET_total++;
    }

    // reset the cursor to the beginning
    if (ViewMovieSET.CursorType > 0) {
        if (!ViewMovieSET.BOF) ViewMovieSET.MoveFirst();
    } else {
        ViewMovieSET.Requery();
    }

    // set the number of rows displayed on this page
    if (ViewMovieSET_numRows < 0 || ViewMovieSET_numRows > ViewMovieSET_total) {
        ViewMovieSET_numRows = ViewMovieSET_total;
    }

    // set the first and last displayed record
    ViewMovieSET_last = Math.min(ViewMovieSET_first + ViewMovieSET_numRows - 1,
ViewMovieSET_total);
    ViewMovieSET_first = Math.min(ViewMovieSET_first, ViewMovieSET_total);
}

```



```

AND M.GenreID=G.GenreID AND M.RatingID=Ra.RatingID AND M.ScreenID=Sc.ScreenID AND
M.SoundID=So.SoundID";
MovieResultSet.CursorType = 0;
MovieResultSet.CursorLocation = 2;
MovieResultSet.LockType = 3;
MovieResultSet.Open();
var MovieResultSet_numRows = 0;
%>
<%
var LocationSET__MMColParam3 = "0";
if(String(Request.QueryString("MovieID")) != "undefined") {
    LocationSET__MMColParam3 = String(Request.QueryString("MovieID"));
}
%>
<%
var LocationSET = Server.CreateObject("ADODB.Recordset");
LocationSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
LocationSET.Source = "SELECT L.LocationName, LM.LMovieID FROM dbo.Location AS L,
dbo.Location_Movie AS LM WHERE L.LocationID=LM.LocationID AND LM.LMovieID="+
LocationSET__MMColParam3.replace("/g, """) + """;
LocationSET.CursorType = 0;
LocationSET.CursorLocation = 2;
LocationSET.LockType = 3;
LocationSET.Open();
var LocationSET_numRows = 0;
%>
<%
var FormatsSET__MMColParam4 = "0";
if(String(Request.QueryString("MovieID")) != "undefined") {
    FormatsSET__MMColParam4 = String(Request.QueryString("MovieID"));
}
%>
<%
var FormatsSET = Server.CreateObject("ADODB.Recordset");
FormatsSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
FormatsSET.Source = "SELECT F.FormatsName, FM.FMovieID FROM dbo.Formats AS F,
dbo.Formats_Movie AS FM WHERE F.FormatsID=FM.FormatsID AND FM.FMovieID="+
FormatsSET__MMColParam4.replace("/g, """) + """;
FormatsSET.CursorType = 0;
FormatsSET.CursorLocation = 2;
FormatsSET.LockType = 3;
FormatsSET.Open();
var FormatsSET_numRows = 0;
%>
<%
var ActorNameSET__MMColParam2 = "0";
if(String(Request.QueryString("MovieID")) != "undefined") {
    ActorNameSET__MMColParam2 = String(Request.QueryString("MovieID"));
}
%>
<%

```

```

var ActorNameSET = Server.CreateObject("ADODB.Recordset");
ActorNameSET.ActiveConnection = MM_MovieCollectionDNSasp_STRING;
ActorNameSET.Source = "SELECT A.ActorFName, A.ActorLName, A.ActorImage, AM.AMovieID,
A.ActorLink FROM dbo.Actor AS A, dbo.Actor_Movie AS AM WHERE A.ActorID=AM.ActorID
AND AM.AMovieID="+ ActorNameSET__MMColParam2.replace(/'/g, "''") + ""';
ActorNameSET.CursorType = 0;
ActorNameSET.CursorLocation = 2;
ActorNameSET.LockType = 3;
ActorNameSET.Open();
var ActorNameSET_numRows = 0;
%>
<%
var Repeat1__numRows = -1;
var Repeat1__index = 0;
ActorNameSET_numRows += Repeat1__numRows;
%>
<%
// *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

// set the record count
var MovieResultSET_total = MovieResultSET.RecordCount;

// set the number of rows displayed on this page
if (MovieResultSET_numRows < 0) { // if repeat region set to all records
    MovieResultSET_numRows = MovieResultSET_total;
} else if (MovieResultSET_numRows == 0) { // if no repeat regions
    MovieResultSET_numRows = 1;
}

// set the first and last displayed record
var MovieResultSET_first = 1;
var MovieResultSET_last = MovieResultSET_first + MovieResultSET_numRows - 1;

// if we have the correct record count, check the other stats
if (MovieResultSET_total != -1) {
    MovieResultSET_numRows = Math.min(MovieResultSET_numRows, MovieResultSET_total);
    MovieResultSET_first = Math.min(MovieResultSET_first, MovieResultSET_total);
    MovieResultSET_last = Math.min(MovieResultSET_last, MovieResultSET_total);
}
%>
<% var MM_paramName = ""; %>
<%
// *** Move To Record and Go To Record: declare variables

var MM_rs = MovieResultSET;
var MM_rsCount = MovieResultSET_total;
var MM_size = MovieResultSET_numRows;
var MM_uniqueCol = "MovieID";
    MM_paramName = "MovieID";
var MM_offset = 0;
var MM_atTotal = false;

```

```

var MM_paramIsDefined = (MM_paramName != "" && String(Request(MM_paramName)) !=
"undefined");
%>
<%
// *** Move To Specific Record: handle detail parameter

if (MM_paramIsDefined && MM_rsCount != 0) {

    // get the value of the parameter
    var param = String(Request(MM_paramName));

    // find the record with the unique column value equal to the parameter value
    for (MM_offset=0; !MM_rs.EOF; MM_offset++) {
        if (MM_rs.Fields.Item(MM_uniqueCol).Value == param) break;
        MM_rs.MoveNext();
    }

    // if not found, set the number of records and reset the cursor
    if (MM_rs.EOF) {
        if (MM_rsCount < 0) MM_rsCount = MM_offset;
        if (MM_size < 0 || MM_size > MM_rsCount) MM_size = MM_rsCount;
        MM_offset = 0;

        // reset the cursor to the beginning
        if (MM_rs.CursorType > 0) {
            if (!MM_rs.BOF) MM_rs.MoveFirst();
        } else {
            MM_rs.Close();
            MM_rs.Open();
        }
    }
}
%>
<%
// *** Move To Record: if we dont know the record count, check the display range

if (MM_rsCount == -1) {

    // walk to the end of the display range for this page
    for (var i=MM_offset; !MM_rs.EOF && (MM_size < 0 || i < MM_offset + MM_size); i++) {
        MM_rs.MoveNext();
    }

    // if we walked off the end of the recordset, set MM_rsCount and MM_size
    if (MM_rs.EOF) {
        MM_rsCount = i;
        if (MM_size < 0 || MM_size > MM_rsCount) MM_size = MM_rsCount;
    }

    // if we walked off the end, set the offset based on page size
    if (MM_rs.EOF && !MM_paramIsDefined) {

```

```

    if ((MM_rsCount % MM_size) != 0) { // last page not a full repeat region
        MM_offset = MM_rsCount - (MM_rsCount % MM_size);
    } else {
        MM_offset = MM_rsCount - MM_size;
    }
}

// reset the cursor to the beginning
if (MM_rs.CursorType > 0) {
    if (!MM_rs.BOF) MM_rs.MoveFirst();
} else {
    MM_rs.Requery();
}

// move the cursor to the selected record
for (var i=0; !MM_rs.EOF && i < MM_offset; i++) {
    MM_rs.MoveNext();
}
}
%>
<%
// *** Move To Record: update recordset stats

// set the first and last displayed record
MovieResultSet_first = MM_offset + 1;
MovieResultSet_last = MM_offset + MM_size;
if (MM_rsCount != -1) {
    MovieResultSet_first = Math.min(MovieResultSet_first, MM_rsCount);
    MovieResultSet_last = Math.min(MovieResultSet_last, MM_rsCount);
}

// set the boolean used by hide region to check if we are on the last record
MM_atTotal = (MM_rsCount != -1 && MM_offset + MM_size >= MM_rsCount);
%>
<html>
<head>
<title>View Movie Results</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body bgcolor="#000000" text="#FFFFFF">
<center>
<p><font color="#FFFFFF" size="5" face="Arial, Helvetica, sans-serif">Movie Results</font></p>
<table border="0">
<tr>
<td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Movie
    Name</font></td>
<td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%=(MovieResultSet.Fields.Item("MovieName").Value)%></font></td>
</tr>
<tr>

```

```

        <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Movie
        Cover </font></td>
    <td><center>
        <font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><A
HREF="<%= (MovieResultSET.Fields.Item("MovieLink").Value)%>"><IMG
SRC="Images/MovieCovers/<%= (MovieResultSET.Fields.Item("MovieImage").Value)%>"
align="absmiddle"></A></font></center></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Movie
    Link</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><A
HREF="<%= (MovieResultSET.Fields.Item("MovieLink").Value)%>"><%= (MovieResultSET.Fields.It
em("MovieName").Value)%> on IMDB</A></font></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">ReleaseYear</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (MovieResultSET.Fields.Item("ReleaseYear").Value)%></font></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Genre</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (MovieResultSET.Fields.Item("GenreName").Value)%></font></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Rating</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (MovieResultSET.Fields.Item("RatingName").Value)%></font></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Screen
    Format</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (MovieResultSET.Fields.Item("ScreenName").Value)%></font></td>
</tr>
<tr>
    <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Sound
    Format</font></td>
    <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (MovieResultSET.Fields.Item("SoundName").Value)%></font></td>
</tr>
<tr>

```

```

        <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Movie
        Format</font></td>
        <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (FormatsSET.Fields.Item("FormatsName").Value)%></font></td>
    </tr>
    <tr>
        <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Movie
        Location</font></td>
        <td><font color="#FFFFFF" face="Arial, Helvetica, sans-
serif"><%= (LocationSET.Fields.Item("LocationName").Value)%></font></td>
    </tr>
    <tr>
        <td align="center" bgcolor="#999999"><font color="#000000" face="Arial, Helvetica, sans-
serif">Actors</font></td>
        <td><table width="100%" border="0" cellpadding="0">
            <tr>
                <% while ((Repeat1 __numRows-- != 0) && (!ActorNameSET.EOF)) { %>
                <td><center>
                    <font color="#FFFFFF" face="Arial, Helvetica, sans-serif"><A
HREF="<%= (ActorNameSET.Fields.Item("ActorLink").Value)%>">"></a><br>
                    <A
HREF="<%= (ActorNameSET.Fields.Item("ActorLink").Value)%>"><%= (ActorNameSET.Fields.Item(
"ActorFName").Value)%> &nbsp;<%= (ActorNameSET.Fields.Item("ActorLName").Value)%></A></f
ont></center></td>
                <%
                Repeat1 __index++;
                ActorNameSET.MoveNext();
            }
            %>
            </tr>
        </table></td>
    </tr>
</table>
</center>
</body>
</html>
<%
MovieResultSET.Close();
%>
<%
LocationSET.Close();
%>
<%
FormatsSET.Close();
%>
<%
ActorNameSET.Close();
%>

```

8 BACK-END DATABASE SQL CODE (SQL SERVER)

8.1 Create Tables

/****** This Script Creates All 14 Tables in the Chris Ballam Movie Database (Formatted for Microsoft SQL Server 2000 *****/

BEGIN TRANSACTION;

CREATE TABLE Release

(ReleaseID INT IDENTITY(1,1) CONSTRAINT Release_pk PRIMARY KEY,
ReleaseYear VARCHAR(50) CONSTRAINT Release_ReleaseYear_nn NOT NULL,
CONSTRAINT Release_ReleaseYear_un UNIQUE(ReleaseYear));

CREATE TABLE Genre

(GenreID INT IDENTITY(1,1) CONSTRAINT Genre_pk PRIMARY KEY,
GenreName VARCHAR(50) CONSTRAINT Genre_GenreName_nn NOT NULL);

CREATE TABLE Rating

(RatingID INT IDENTITY(1,1) CONSTRAINT Rating_pk PRIMARY KEY,
RatingName VARCHAR(50) CONSTRAINT Rating_RatingName_nn NOT NULL);

CREATE TABLE Screen

(ScreenID INT IDENTITY(1,1) CONSTRAINT Screen_pk PRIMARY KEY,
ScreenName VARCHAR(50) CONSTRAINT Screen_ScreenName_nn NOT
NULL);

CREATE TABLE Sound

(SoundID INT IDENTITY(1,1) CONSTRAINT Sound_pk PRIMARY KEY,
SoundName VARCHAR(50) CONSTRAINT Sound_SoundName_nn NOT NULL);

CREATE TABLE Borrower

(BorrowerID INT IDENTITY(1,1) CONSTRAINT Borrower_pk PRIMARY KEY,
BorrowerFName VARCHAR(50) CONSTRAINT Borrower_BorrowerFName_nn NOT NULL,
BorrowerLName VARCHAR(50) CONSTRAINT Borrower_BorrowerLName_nn NOT NULL,
BorrowerPhone VARCHAR(50),
BorrowerEmail VARCHAR(50));

CREATE TABLE Actor

(ActorID INT IDENTITY(1,1) CONSTRAINT Actor_pk PRIMARY KEY,
ActorFName VARCHAR(50) CONSTRAINT Actor_ActorFName_nn NOT NULL,
ActorLName VARCHAR(50) CONSTRAINT Actor_ActorLName_nn NOT NULL,
ActorLink VARCHAR(100) CONSTRAINT Actor_ActorLink_nn NOT NULL,
ActorImage VARCHAR(50));

CREATE TABLE Formats

(FormatsID INT IDENTITY(1,1) CONSTRAINT Formats_pk PRIMARY KEY,
FormatsName VARCHAR(50) CONSTRAINT Formats_FormatsName_nn NOT NULL);

CREATE TABLE Location

(LocationID INT IDENTITY(1,1) CONSTRAINT Location_pk PRIMARY KEY,
LocationName VARCHAR(50) CONSTRAINT Location_LocationName_nn NOT NULL);

CREATE TABLE Movie

(MovieID INT IDENTITY(1,1) CONSTRAINT Movie_pk PRIMARY KEY,
MovieName VARCHAR(100) CONSTRAINT Movie_MovieName_nn NOT NULL,
MovieImage VARCHAR(100) CONSTRAINT Movie_MovieImage_nn NOT NULL,
MovieLink VARCHAR(100) CONSTRAINT Movie_MovieLink_nn NOT NULL,
ReleaseID INT CONSTRAINT Movie_ReleaseID_nn NOT NULL,
CONSTRAINT Movie_ReleaseID_fk FOREIGN KEY(ReleaseID) REFERENCES Release,
GenreID INT CONSTRAINT Movie_GenreID_nn NOT NULL,
CONSTRAINT Movie_GenreID_fk FOREIGN KEY(GenreID) REFERENCES Genre,
RatingID INT CONSTRAINT Movie_RatingID_nn NOT NULL,
CONSTRAINT Movie_RatingID_fk FOREIGN KEY(RatingID) REFERENCES Rating,
ScreenID INT CONSTRAINT Movie_ScreenID_nn NOT NULL,
CONSTRAINT Movie_ScreenID_fk FOREIGN KEY(ScreenID) REFERENCES Screen,
SoundID INT CONSTRAINT Movie_SoundID_nn NOT NULL,
CONSTRAINT Movie_SoundID_fk FOREIGN KEY(SoundID) REFERENCES Sound);

CREATE TABLE Borrowed_Movie

(BorrowedMovieID INT IDENTITY(1,1) CONSTRAINT BorrowedMovie_pk PRIMARY KEY,
BorrowedMovieDate DATETIME CONSTRAINT BorrowedMovie_BorrowedMovieDate_nn
NOT NULL,
BMovieID INT CONSTRAINT BorrowedMovie_BMovieID_nn NOT NULL,
CONSTRAINT BorrowedMovie_BMovieID_fk FOREIGN KEY(BMovieID) REFERENCES
Movie(MovieID),
BorrowerID INT CONSTRAINT BorrowedMovie_BorrowerID_nn NOT NULL,
CONSTRAINT BorrowedMovie_BorrowerID_fk FOREIGN KEY(BorrowerID) REFERENCES
Borrower);

CREATE TABLE Actor_Movie

(ActorMovieID INT IDENTITY(1,1) CONSTRAINT ActorMovie_pk PRIMARY KEY,
ActorID INT CONSTRAINT ActorMovie_ActorID_nn NOT NULL,
CONSTRAINT ActorMovie_ActorID_fk FOREIGN KEY(ActorID) REFERENCES Actor,
AMovieID INT CONSTRAINT ActorMovie_AMovieID_nn NOT NULL,
CONSTRAINT ActorMovie_AMovieID_fk FOREIGN KEY(AMovieID) REFERENCES
Movie(MovieID));

CREATE TABLE Formats_Movie

(FormatsMovieID INT IDENTITY(1,1) CONSTRAINT FormatsMovie_pk PRIMARY KEY,
FormatsID INT CONSTRAINT FormatsMovie_FormatsID_nn NOT NULL,
CONSTRAINT FormatsMovie_FormatID_fk FOREIGN KEY(FormatsID) REFERENCES
Formats,
FMovieID INT CONSTRAINT FormatsMovie_FMovieID_nn NOT NULL,
CONSTRAINT FormatsMovie_FMovieID_fk FOREIGN KEY(FMovieID) REFERENCES
Movie(MovieID));

CREATE TABLE Location_Movie

(LocationMovieID INT IDENTITY(1,1) CONSTRAINT LocationMovie_pk PRIMARY KEY,
LocationID INT CONSTRAINT LocationMovie_LocationID_nn NOT NULL,

```

        CONSTRAINT LocationMovie_LocationID_fk FOREIGN KEY(LocationID) REFERENCES
Location,
        LMovieID          INT    CONSTRAINT LocationMovie_LMovieID_nn NOT NULL,
        CONSTRAINT LocationMovie_LMovieID_fk FOREIGN KEY(LMovieID) REFERENCES
Movie(MovieID));

COMMIT TRANSACTION;

```

8.2 Insert Set Data

```

/***** This Script Inserts Set Data into 7 Tables (Formatted for Microsoft SQL Server 2000)
*****/

```

```

BEGIN TRANSACTION;

```

```

/***** This Script Adds Release Years to the Database *****/

```

```

INSERT INTO Release(ReleaseYear) VALUES('1894');
INSERT INTO Release(ReleaseYear) VALUES('1895');
INSERT INTO Release(ReleaseYear) VALUES('1896');
INSERT INTO Release(ReleaseYear) VALUES('1897');
INSERT INTO Release(ReleaseYear) VALUES('1898');
INSERT INTO Release(ReleaseYear) VALUES('1899');
INSERT INTO Release(ReleaseYear) VALUES('1900');
INSERT INTO Release(ReleaseYear) VALUES('1901');
INSERT INTO Release(ReleaseYear) VALUES('1902');
INSERT INTO Release(ReleaseYear) VALUES('1903');
INSERT INTO Release(ReleaseYear) VALUES('1904');
INSERT INTO Release(ReleaseYear) VALUES('1905');
INSERT INTO Release(ReleaseYear) VALUES('1906');
INSERT INTO Release(ReleaseYear) VALUES('1907');
INSERT INTO Release(ReleaseYear) VALUES('1908');
INSERT INTO Release(ReleaseYear) VALUES('1909');
INSERT INTO Release(ReleaseYear) VALUES('1910');
INSERT INTO Release(ReleaseYear) VALUES('1911');
INSERT INTO Release(ReleaseYear) VALUES('1912');
INSERT INTO Release(ReleaseYear) VALUES('1913');
INSERT INTO Release(ReleaseYear) VALUES('1914');
INSERT INTO Release(ReleaseYear) VALUES('1915');
INSERT INTO Release(ReleaseYear) VALUES('1916');
INSERT INTO Release(ReleaseYear) VALUES('1917');
INSERT INTO Release(ReleaseYear) VALUES('1918');
INSERT INTO Release(ReleaseYear) VALUES('1919');
INSERT INTO Release(ReleaseYear) VALUES('1920');
INSERT INTO Release(ReleaseYear) VALUES('1921');
INSERT INTO Release(ReleaseYear) VALUES('1922');
INSERT INTO Release(ReleaseYear) VALUES('1923');
INSERT INTO Release(ReleaseYear) VALUES('1924');
INSERT INTO Release(ReleaseYear) VALUES('1925');
INSERT INTO Release(ReleaseYear) VALUES('1926');
INSERT INTO Release(ReleaseYear) VALUES('1927');
INSERT INTO Release(ReleaseYear) VALUES('1928');
INSERT INTO Release(ReleaseYear) VALUES('1929');

```

[illegible]

[illegible]

```

INSERT INTO Release(ReleaseYear) VALUES('2032');
INSERT INTO Release(ReleaseYear) VALUES('2033');
INSERT INTO Release(ReleaseYear) VALUES('2034');
INSERT INTO Release(ReleaseYear) VALUES('2035');
INSERT INTO Release(ReleaseYear) VALUES('2036');
INSERT INTO Release(ReleaseYear) VALUES('2037');
INSERT INTO Release(ReleaseYear) VALUES('2038');
INSERT INTO Release(ReleaseYear) VALUES('2039');
INSERT INTO Release(ReleaseYear) VALUES('2040');
INSERT INTO Release(ReleaseYear) VALUES('2041');
INSERT INTO Release(ReleaseYear) VALUES('2042');
INSERT INTO Release(ReleaseYear) VALUES('2043');
INSERT INTO Release(ReleaseYear) VALUES('2044');
INSERT INTO Release(ReleaseYear) VALUES('2045');
INSERT INTO Release(ReleaseYear) VALUES('2046');
INSERT INTO Release(ReleaseYear) VALUES('2047');
INSERT INTO Release(ReleaseYear) VALUES('2048');
INSERT INTO Release(ReleaseYear) VALUES('2049');
INSERT INTO Release(ReleaseYear) VALUES('2050');

```

/****** This Script Adds All the Genres to the Database *****/

```

INSERT INTO Genre(GenreName) VALUES('Action / Adventure');
INSERT INTO Genre(GenreName) VALUES('Animation');
INSERT INTO Genre(GenreName) VALUES('Anime');
INSERT INTO Genre(GenreName) VALUES('Comedy');
INSERT INTO Genre(GenreName) VALUES('Concert / Music');
INSERT INTO Genre(GenreName) VALUES('Dance / Ballet');
INSERT INTO Genre(GenreName) VALUES('Documentary');
INSERT INTO Genre(GenreName) VALUES('Drama');
INSERT INTO Genre(GenreName) VALUES('Exercise');
INSERT INTO Genre(GenreName) VALUES('Family');
INSERT INTO Genre(GenreName) VALUES('Foreign');
INSERT INTO Genre(GenreName) VALUES('Horror');
INSERT INTO Genre(GenreName) VALUES('Karaoke');
INSERT INTO Genre(GenreName) VALUES('Mature');
INSERT INTO Genre(GenreName) VALUES('Musical');
INSERT INTO Genre(GenreName) VALUES('Mystery / Suspense');
INSERT INTO Genre(GenreName) VALUES('Opera');
INSERT INTO Genre(GenreName) VALUES('Other');
INSERT INTO Genre(GenreName) VALUES('Sci-Fi / Fantasy');
INSERT INTO Genre(GenreName) VALUES('Software');
INSERT INTO Genre(GenreName) VALUES('Special Interest');
INSERT INTO Genre(GenreName) VALUES('Sports');
INSERT INTO Genre(GenreName) VALUES('TV Classics');
INSERT INTO Genre(GenreName) VALUES('Western');

```

/****** This Script Adds All the Movie Ratings to the Database *****/

```

INSERT INTO Rating(RatingName) VALUES('G');
INSERT INTO Rating(RatingName) VALUES('NR');
INSERT INTO Rating(RatingName) VALUES('PG');
INSERT INTO Rating(RatingName) VALUES('PG-13');

```

```
INSERT INTO Rating(RatingName) VALUES('R');
INSERT INTO Rating(RatingName) VALUES('NC-17');
```

```
/****** This Script Adds All the Screen Sizes to the Database *****/
```

```
INSERT INTO Screen(ScreenName) VALUES('4:3');
INSERT INTO Screen(ScreenName) VALUES('4:3 / LBX');
INSERT INTO Screen(ScreenName) VALUES('4:3 / LBX / 16:9');
INSERT INTO Screen(ScreenName) VALUES('LBX');
INSERT INTO Screen(ScreenName) VALUES('LBX / 16:9');
INSERT INTO Screen(ScreenName) VALUES('16:9');
```

```
/****** This Script Adds All the Sound Formats to the Database *****/
```

```
INSERT INTO Sound(SoundName) VALUES('1.0');
INSERT INTO Sound(SoundName) VALUES('2.0');
INSERT INTO Sound(SoundName) VALUES('5.1');
INSERT INTO Sound(SoundName) VALUES('DTS');
INSERT INTO Sound(SoundName) VALUES('DTS / 5.1');
```

```
/****** This Script Adds All the Movie Formats to the Database *****/
```

```
INSERT INTO Formats(FormatsName) VALUES('DVD');
INSERT INTO Formats(FormatsName) VALUES('VCD (ASF)');
INSERT INTO Formats(FormatsName) VALUES('VCD (ASF - DVD)');
INSERT INTO Formats(FormatsName) VALUES('VCD (ASF - Screener)');
INSERT INTO Formats(FormatsName) VALUES('VCD (ASX)');
INSERT INTO Formats(FormatsName) VALUES('VCD (AVI)');
INSERT INTO Formats(FormatsName) VALUES('VCD (AVI - DivX)');
INSERT INTO Formats(FormatsName) VALUES('VCD (AVI - DivX DVD)');
INSERT INTO Formats(FormatsName) VALUES('VCD (AVI - Screener)');
INSERT INTO Formats(FormatsName) VALUES('VCD (MOV)');
INSERT INTO Formats(FormatsName) VALUES('VCD (MPG)');
INSERT INTO Formats(FormatsName) VALUES('VCD (MPG - Screener)');
INSERT INTO Formats(FormatsName) VALUES('VCD (RM)');
INSERT INTO Formats(FormatsName) VALUES('VCD (VCD)');
INSERT INTO Formats(FormatsName) VALUES('VHS');
```

```
/****** This Script Adds Movie Locations to the Database *****/
```

```
INSERT INTO Location(LocationName) VALUES('1 CD');
INSERT INTO Location(LocationName) VALUES('2 CD');
INSERT INTO Location(LocationName) VALUES('3 CD');
INSERT INTO Location(LocationName) VALUES('4 CD');
INSERT INTO Location(LocationName) VALUES('DVD');
INSERT INTO Location(LocationName) VALUES('VHS');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #1');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #2');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #3');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #4');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #5');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #6');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #7');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #8');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #9');
```

[illegible]

[illegible]

[illegible]

```

INSERT INTO Location(LocationName) VALUES('Full MovieZ #163');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #164');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #165');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #166');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #167');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #168');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #169');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #170');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #171');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #172');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #173');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #174');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #175');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #176');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #177');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #178');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #179');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #180');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #181');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #182');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #183');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #184');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #185');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #186');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #187');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #188');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #189');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #190');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #191');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #192');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #193');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #194');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #195');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #196');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #197');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #198');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #199');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #200');
INSERT INTO Location(LocationName) VALUES('Full MovieZ #201');

```

```
COMMIT TRANSACTION;
```

8.3 Insert Sample Movies

```
/****** This Script Adds 4 Movies to the Database (Formatted for Microsoft SQL Server 2000)
```

```
*****/
```

```
BEGIN TRANSACTION;
```

```

INSERT INTO Movie(MovieName, MovieImage, MovieLink, ReleaseID, GenreID, RatingID,
ScreenID, SoundID) VALUES('Crouching Tiger Hidden Dragon: Special Edition', 'Crouching Tiger
Hidden Dragon.jpg', 'http://us.imdb.com/Title?0190332', 107, 11, 4, 5, 3);
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Yun-Fat', 'Chow',
'http://us.imdb.com/Name?Chow,+Yun-Fat', 'Yun-FatChow.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Michelle', 'Yeoh',
'http://us.imdb.com/Name?Yeoh,+Michelle', 'MichelleYeoh.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Ziyi', 'Zhang',
'http://us.imdb.com/Name?Zhang,+Ziyi', 'ZiyiZhang.jpg');
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(1, 1);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(2, 1);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(3, 1);
INSERT INTO Formats_Movie(FormatsID, FMovieID) VALUES(1, 1);
INSERT INTO Location_Movie(LocationID, LMovieID) VALUES(5, 1);

```

```

INSERT INTO Movie(MovieName, MovieImage, MovieLink, ReleaseID, GenreID, RatingID,
ScreenID, SoundID) VALUES('Frequency', 'Frequency.jpg', 'http://us.imdb.com/Title?0186151', 107,
19, 4, 5, 3);
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Dennis', 'Quaid',
'http://us.imdb.com/Name?Quaid,+Dennis', 'DennisQuaid.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('James',
'Caviezel', 'http://us.imdb.com/Name?Caviezel,+James', 'JamesCaviezel.jpg');
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(4, 2);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(5, 2);
INSERT INTO Formats_Movie(FormatsID, FMovieID) VALUES(1, 2);
INSERT INTO Location_Movie(LocationID, LMovieID) VALUES(5, 2);

```

```

INSERT INTO Movie(MovieName, MovieImage, MovieLink, ReleaseID, GenreID, RatingID,
ScreenID, SoundID) VALUES('Mission Impossible 2', 'MissionImpossible2.jpg',
'http://us.imdb.com/Title?0120755', 107, 1, 4, 5, 3);
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Tom', 'Cruise',
'http://us.imdb.com/Name?Cruise,+Tom', 'TomCruise.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Dougray', 'Scott',
'http://us.imdb.com/Name?Scott,+Dougray', 'DougrayScott.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Thandie',
'Newton', 'http://us.imdb.com/Name?Newton,+Thandie', 'ThandieNewton.jpg');
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(6, 3);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(7, 3);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(8, 3);
INSERT INTO Formats_Movie(FormatsID, FMovieID) VALUES(1, 3);
INSERT INTO Location_Movie(LocationID, LMovieID) VALUES(5, 3);

```

```

INSERT INTO Movie(MovieName, MovieImage, MovieLink, ReleaseID, GenreID, RatingID,
ScreenID, SoundID) VALUES('Mission Impossible', 'MissionImpossible.jpg',
'http://us.imdb.com/Title?0117060', 103, 1, 4, 2, 3);
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Jon', 'Voight',
'http://us.imdb.com/Name?Voight,+Jon', 'JonVoight.jpg');
INSERT INTO Actor(ActorFName, ActorLName, ActorLink, ActorImage) VALUES('Emmanuelle',
'Béart', 'http://us.imdb.com/Name?B%E9art,+Emmanuelle', 'EmmanuelleBeart.jpg');
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(6, 4);
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(9, 4);

```

```
INSERT INTO Actor_Movie(ActorID, AMovieID) VALUES(10, 4);
INSERT INTO Formats_Movie(FormatsID, FMovieID) VALUES(1, 4);
INSERT INTO Location_Movie(LocationID, LMovieID) VALUES(5, 4);
```

```
COMMIT TRANSACTION;
```

8.4 Insert Sample Borrowers

```
/****** This Script Adds Borrowers to the Database (Formatted for Microsoft SQL Server 2000)
*****/
```

```
BEGIN TRANSACTION;
```

```
INSERT INTO Borrower(BorrowerID, BorrowerFName, BorrowerLName, BorrowerPhone,
BorrowerEmail) VALUES(1, 'John', 'Doe', '(123)456-7890', 'johndoe@example.com');
```

```
INSERT INTO Borrower(BorrowerID, BorrowerFName, BorrowerLName, BorrowerPhone,
BorrowerEmail) VALUES(2, 'Fake', 'Person', '(123)456-7890', 'fakeperson@example.com');
```

```
COMMIT TRANSACTION;
```

8.5 Drop All Tables

```
/****** This Script Drops All 14 Tables in Chris Ballam's Movie Database (Formatted for Microsoft
SQL Server 2000) *****/
```

```
BEGIN TRANSACTION;
```

```
DROP TABLE Borrowed_Movie;
DROP TABLE Actor_Movie;
DROP TABLE Formats_Movie;
DROP TABLE Location_Movie;
DROP TABLE Movie;
DROP TABLE Borrower;
DROP TABLE Actor;
DROP TABLE Formats;
DROP TABLE Location;
DROP TABLE Sound;
DROP TABLE Screen;
DROP TABLE Rating;
DROP TABLE Genre;
DROP TABLE Release;
```

```
COMMIT TRANSACTION;
```

9 REFERENCES

- Clowes, Shawn. PHP.net (Accessed December 2001). "PHP 4.1.0 Release Announcement" December 2001. Posted at: http://www.php.net/release_4_1_0.php. PHP.net: WWW.
- Culbert, Brad. Webmaster's Library (Accessed December 2001). "Interview with Tobias Ratschiller." June 2001. Posted at: <http://www.webmasterslibrary.com/interviews/content.jsp?phpwizard>. Webmaster's Library.com: WWW.
- Dragan, Richard V. PC Magazine Online (Accessed November 2001). "JSP and Tomcat 3.2." May 2001. Posted at: http://www.pcmag.com/print_article.0,3048,a%3D2851,00.asp. PC Magazine Online: WWW.
- Desai, Gautam. Network Computing (Accessed December 2001). "Web Application Servers Come of Age." July 2001. Posted at: <http://www.networkcomputing.com/1215/1215f4.html>. NetworkComputing.com: WWW.
- Dyck, Timothy. PC Magazine Online (Accessed December 2001). "Web Server Brains." May 2001. Posted at: http://www.pcmag.com/print_article.0,3048,a%3D2851,00.asp. PC Magazine Online: WWW.
- Garris, John. PC Magazine Online (Accessed December 2001). "Web Application Servers." September 1998. Posted at: <http://www.zdnet.com/pcmag/pctech/content/17/15/tf1715.001.html>. PC Magazine Online: WWW.
- Gunnerson, Gary. PC Magazine Online (Accessed November 2001). "WebSphere Application Server, Advanced Edition 3.5." May 2001. Posted at: http://www.pcmag.com/print_article.0,3048,a%3D2851,00.asp. PC Magazine Online: WWW.
- Jepson, Brian. PC Magazine Online (Accessed November 2001). "It Takes a Database." July 2001. Posted at: <http://www.pcmag.com/article/0,2997,s%253D1487%2526a%253D5626,00.asp>. PC Magazine Online: WWW.
- Lam, John. PC Magazine Online (Accessed December 2001). "Microsoft IIS, COM+, and ASP." May 2001. Posted at: http://www.pcmag.com/print_article.0,3048,a%3D2851,00.asp. PC Magazine Online: WWW.
- Neo FAQ. Macromedia.com (Accessed December 2001). "Neo FAQ." Posted at: <http://www.macromedia.com/software/coldfusion/resources/neo/faq/>. Macromedia.com: WWW.
- Pelegri-Llopart, Edwardo. Java Live (Accessed November 2001). "JavaServer Pages™ Technology and JSP™ tag Libraries." October 2001. Posted at: <http://developer.java.sun.com/developer/community/chat/JavaLive/2001/j11023.html>. Java Live: WWW.
- Phillips, Chuck. CNET News.com (Accessed December 2001). "Java application servers: The next OS?" July 2001. Posted at: <http://news.cnet.com/news/0-1273-210-6618787-1.html?tag=archlnk>. CNET.com Tech News: WWW.

- Press Release. Macromedia.com (Accessed December 2001). "Press Release." Posted at: http://www.macromedia.com/macromedia/proom/pr/2001/index_neo_partners.fhtml. Macromedia.com: WWW.
- Sergey (Accessed December 2001). "Server-Side technologies (Application Servers)." February 2001. Posted at: http://www.sergey.com/web_course/part_11.html. Sergey.com: WWW.
- Stoeckel, John. "Back to the Future: Plan Ahead for e-business by Exploring Web Application Servers." December 1999. AS/400. Update from PartnerWorld for Developers. IBM e-business.
- Trachtenberg, Adam. CNET Builder.com (Accessed November 2001) "Introduction to PHP." August 1999. Posted at: <http://builder.cnet.com/webbuilding/pages/Programming/PHPIntro/>. CNET.com Web Building: WWW.
- Walker, Michael. "Instant PHP 4." October 2001. Osborne/McGraw-Hill. 550 Pages.
- Why ASP.NET? Microsoft ASP.NET (Accessed December 2001). "Why ASP.NET?" Posted at: <http://www.asp.net/whitepaper/whyaspnet.aspx>. Microsoft ASP.NET: WWW.