Evaluation of Two Web Development Technologies
Implementation of an online shopping store

Master Thesis, Computer Science

Author:
Muhammad Umair Zahid(mzd09001@student.mdh.se)
Mehdi Mohammed Hasan Rahi(mri09003@student.mdh.se)

Supervisor/Examiner:
Frank Lüders
frank.luders@mdh.se
Abstract

In the thesis we have evaluate the differences between two web development technologies by developing the same two applications with same functional requirement in different two technologies, Microsoft Visual Studio 2010 and Netbeans IDE 6.8. We use the Active Server Pages (ASP.NET) with C# in the Visual Studio 2010 and Java Server Pages (JSP) with Servlet in the NetBeans IDE 6.8. We have developed an online shopping store with some features for Admin and Client. We have found that NetBeans with JSP is faster in execution of the code as compare to Visual Studio with ASP.NET when we update any code because JSP compile the code at runtime. Meanwhile we have also found that ASP.NET is faster for development procedure because ASP.NET provides many built-in tools, controls and functions while as in JSP there is not so much built-in help to generate the code automatically. We have also face that Visual Studio with ASP.NET is easier to learn as compared to NetBeans with JSP because ASP.NET has strong and complete documentation inform of tutorials, books and examples.
Contents

1. Introduction........................................................................................................................................ 5
   1.1. Background .................................................................................................................................. 5
2. .NET with Microsoft Visual Studio 2010 .............................................................................................. 6
   2.1. .NET Framework .......................................................................................................................... 6
   2.2. Visual Studio 2010 Technology: .................................................................................................... 8
3. Java with NetBeans............................................................................................................................. 10
   3.1. Servlet Technology ..................................................................................................................... 10
   3.2. Java Server Pages (JSP) ............................................................................................................. 14
4. Application Design............................................................................................................................. 15
   4.1. Database Design ........................................................................................................................... 16
   4.2. Implementation Design ................................................................................................................ 16
   4.3. User Interface ................................................................................................................................ 17
5. Implementation Application Steps in .NET c# .............................................................................. 23
   5.1. Install VS 2010 and SQL Server 2008 .......................................................................................... 23
   5.2. Create Database & Connection .................................................................................................... 23
   5.3. Create Web Application Project .................................................................................................. 24
   5.4. Create and Implement Classes ...................................................................................................... 24
   5.5. Create & implement Master pages ............................................................................................... 26
   5.6. ASP.NET Content Pages .............................................................................................................. 27
6. Implementation Application Steps in JSP Netbeans ...................................................................... 29
   6.1. Install NetBeans IDE6.8 and MySQL .......................................................................................... 29
   6.2. Create Database & Connection .................................................................................................... 30
   6.3. Create Web Application Project: ............................................................................................... 31
6.4. Create and Implement Classes ................................................................. 32
6.5. Application Pages in JSP: ................................................................. 34
7. Experience of Implementation in Both Platform ........................................... 37
8. Conclusion & Future Work ......................................................................... 38
9. References ............................................................................................... 39

Appendix A: Visual Studio 2010 Implementation ........................................... 40
  A.1 Implement user class ........................................................................... 40
  A.2 Implement client class ......................................................................... 41
  A.3 Implement Product class ..................................................................... 42
  A.4 Implement Product_Category class .................................................... 47
  A.5 Implement Web.Config ........................................................................ 51

Appendix B: NetBeans IDE Implementation ............................................... 52
  B.1 Initialize Product_Category class ....................................................... 52
  B.2 Implement method getCategoryProduct insideProduct_Category class .................................................. 52
  B.3 Implement method GetallCategory ..................................................... 53
  B.4 Implement menu JSP page ................................................................. 54
  B.5 The structure of Servlet....................................................................... 56
1. Introduction
In this thesis we developed the two web applications in different two platforms of different languages with same functional requirements. Further more we have tried to compare the both platform in case of which platform is more user friendly, reliable, fast development, and more secure. The web application is about online shopping store that contains the several functional requirements. The application contains the 3 panel which are Guest panel, User Panel and Administrator panel that have the different rights and functionalities perceptively. In the guest panel guest can search, view and add product in shopping cart and guest can be registered by giving some compulsory information. In the User panel User can search, view, add product in shopping cart and as well as customer can also purchase the selected products that’s must be in Shopping cart. In the administrative panel, administrative can Add categories, Update categories, add products and update products.

We assigned two different platforms for these web applications that are are Microsoft Visual Studio 2010 and Sun Microsystems Java (Net Beans IDE 6.8). We selected two technologies Active Server Pages (ASP.NET) and Java Server Pages (JSP) respectively for development in both platforms. Both languages are very popular and efficient for dynamic web applications. Both languages have the Several Versions and the versions came with the passage of time. We have described the history of the both languages in Background Section. This Section will describe the history of the both languages.

1.1. Background
ASP.NET came from Classis ASP. In Classic ASP there is a big problem in separation of the code and design that Microsoft resolves in ASP+ version. ASP+ was the latest version of the classis ASP. Later on Microsoft give name it as ASP.NET. Microsoft has launched several versions with some improvement and fixing bugs in earlier version. ASP.NET version 1.0 came in start and then version ASP.NET 2.0 came and then ASP.NET version 3.0 came and now the latest version of the ASP.NET is 4.0 that contains the lots of modification in earlier versions and lots of new user built-in functionality .ASP.NET 2010 is complete and strong platform for built-in dynamic websites.[1]
Servlet was created by Sun Microsystem in June 1997. It came with version 1.0. JSP was also introduced not to replace Servlet but to extend the functionality and help developers. It introduced in 1999 by Sun Microsystem. While ASP and PHP was providing enough support then Sun decided to introduce JSP to give enough support to developer for web.

In coming parts of the report we have described about the Visual Studio 2010 and NetBeans Technologies in detail how they work and what are the features of both technologies. Then we have described the implementation phase of the application in both technologies from start to finish. Then we have described our experience in both technologies that how we found both technologies during the implementation phase and then we have described the conclusion and future work of our thesis topic in the report. We have also mentioned the Reference section after the conclusion part that contains the some links of the sources where from we got the information related our thesis title. At the end of the report we have given the Appendix section that contains the source code of both technologies.

2. .NET with Microsoft Visual Studio 2010

2.1. .NET Framework
What is the .NET? What is the meant of .NET? Every people have different thinking about the .NET but according to the official .NET Framework is Microsoft platform for the XML Web Services. .NET framework came especially for the Interment Application in mind as well as traditional desktop applications. Internet has spread around the world from many last years and interment plays important role in human life now days. Before .NET Microsoft Component Object Model (COM) take several years to develop before interment popularity. COM was developed especially for traditional desktop applications but .NET covers all application especially internet application and Microsoft built the .NET from the scratch to the final product that support all applications.

The .NET framework consists of languages, technologies and tools that enable to the developer to make the applications language independent and .NET framework also enable the communication between different applications. The .NET framework consists of 2 parts called Common Language Runtime(CLR) and Base class libraries(BCL) .Both
part of the framework are important in development phases and services of the application. Is .NET framework looking reliable, secure and efficient? Is .NET framework provides the Portability? [2]

The .NET framework contains several pre built-in libraries, common functions, file, XML and APIs for all the languages by using Base Class Library (BCL) that the developer can makes the ease to code the application. The .NET framework also provides the highly security to all applications by controlling the buffer overflows that can happen by malicious software. The .NET framework also provides the high quality Graphical User interface for the applications like Silver Light and Windows Presentation foundation framework that help to the developer to make the application in 2-d and 3-d. Silver light framework is mostly used to make the high quality media interactive applications. Both frameworks are the latest technologies in the .NET Microsoft platform. The .NET framework is complete and powerful platform to make all kind of the applications in less time and in less cost. Microsoft launches the several .NET frameworks with the passage of time. The first Microsoft framework was 1.0 and the latest version is 4.0 in Visual Studio 2010. The .NET framework provides the interoperability functionality to the system for the communication between old and new applications by using the libraries. In .NET framework all programs run under the control of Common Runtime Engine (CLR). The .NET execution engine (CLR) makes the .NET framework more reliable and secure by managing the memory, security and exception during the execution of the program. The .NET framework support language independence platform by providing the Common Type System. The .NET framework also provides the portability because the .NET framework architecture construct like .NET framework place on the top layer of the operating System. That’s why the third party companies can run .NET framework and its languages on the other platforms.

There was a vision came to connect all devices together and have communication between them from computing industry. Microsoft gave the very much importance on this new vision and success to develop the solution in .NET framework that’s support to connect all household and commercial devices together and communicate each other by using the XML web services and SOAP. [2, 3, 4]
2.2. **Visual Studio 2010 Technology:**
There are many new features and improvement introduce in Microsoft Visual Studio 2010 with framework 4.0. We have described the short overview about the Visual Studio 2010 in our report. Visual Studio 2010 supports the compatible with the older version like framework 3.5, 3.0, 2.0 and many versions can run at same process. Last Visual Studio framework 3.5 also compatible with older versions but there was problem in some cases, for example if a function called the COM component that’s is also written in same function but with different version of .NET then there was chance to occur problem. The VS 2010 solve this problem and in this version both assemblies use their own version in which they are developed.

In Visual Studio 2010 there are new platform in Ajax, Silver light and the ASP.NET. In Visual Studio 2010 desktop application also has improved by introducing the Work Presentation Foundation (WPF). In Web Development platform ASP.NET also has
improved by introducing ASP.NET MVC that’s helpful in testing. In Parallel programming there is also introduce many new libraries that is helpful to improve performance on multi core processor. Visual Studio 2010 introduces improvement for windows Azure that is uses to control and deploy the different devices together in Cloud form. In visual Studio 2010 Microsoft has major improvement in Visualization and graphically looks of the application using WPF. In visual studio 2010 there improvement introduce of separate code from of presentation layer to business logic layer. Zooming is new feature that introduce in VS 2010, zoom in is use to show code bigger and zoom out is use for to show the code smaller to the audience. [5]

In Visual Studio 2010 developer not need to buy or install separate tool for UML designing. Visual Studio 2010 has built-in sport of UML designing for the developer. In Visual Studio 2010 there are built-in codes libraries that developer not need to code the application, just developer need to use but need to make documentation of the application. This feature makes the .NET developer more ease and fast to develop the application. In visual studio 2010 developer can make the Architecture of the application using the code, by using this feature new developer can understand the application structure. Visual studio 2010 provides the tool for the testing purpose to make the test cases and execute and save the test cases. This feature is very helpful for the tester and developer of the application. .NET framework 4.0 also provides the deployment tools. In framework 2010 .NET introduce new language F# that provides the functional programming to the .NET developer. This language is specially used for the calculation and mathematical expressions. F# is also helpful in concurrent programming with high performance.

There are several other improvements in .NET 2010 framework e.g. in Debugging, Threading, Garbage Collector, Data Access, Redevelopment, SharePoint development, Visual C++, C# and Visual Basic languages. The main improvement in c# is introduce of dynamic type that makes the c# more dynamic to handle errors and exception at runtime and main advantage of the dynamic type to make C# interoperability with COM and other languages. There are several modifications in web development (ASP.NET 4.0), for
example introduces the JQuery, ASP.NET MVC, improvement in performance and visualization. [6, 7]

3. **Java with NetBeans**

NetBeans is an integrated developed environment (IDE) for developers. It is developed by Sun Microsystems [8, 9]. It is entirely written in java. It offers wide range of tools to build professional applications. It is easy to use. Anyone can create enterprise, desktop, web or mobile applications here. IDE contains numbers of tools and support for developers. It consists of source code editor, compiler, debugging support, window system for viewing projects and project artifacts and integration with other commonly-used services. IDE helps developers to reduce their work.

IDE automatically creates build ant script which handles compile, clean, deploy the project. When a developer run his project from IDE, it is compiled, deployed and open in default browser by IDE.

NetBeans IDE 6.8 is the first IDE with comes with full features of Java EE 6. It also offers numbers of improved features for JSF2.0/ Facelets, Java Persistence 2.0, EJB 3.1 and using EJB web applications. The other special features include RESTful web services, GlassFish v3, JavaFX SDK 1.2.1 and PHP with new PHP 5.3 release .etc. Developer can host their open source projects with ProjectKenai which is an environment for hosting open source project. The IDE also has the feature of full JIRA support, improved instant messenger.

Java EE is widely used platform for java server programming. It is used for developing web applications. It is different from Java Standard Edition Platform (Java SE). It contains libraries which gives extra functionalities to create fault tolerant, distributed, multi-tier software which runs on application server. Java EE provides a framework for different services on Java platform.

3.1. **Servlet Technology**

Servlet is the core of web application development in java. It is the most important underlying technology for web application development. Servlets work with JSP (Java
Servlet technology is very important to develop a more efficient and effective JSP application.

Dynamic web programming was possible for Common Gateway Interface (CGI) technology. Through common gateway interface HTTP request process by external program. After process the request it sends back to the web server. Web server forwards it to the client browser. Perl was popular language to write CGI programs before.

With the time internet becomes popular and people visiting web site increased day by day. At the beginning it was CGI technology. There are numbers of flaw inside CGI technology. The flaw was with each client request it creates a new process which eats a lot of memory. So the performance was not efficient to satisfy users. Soon new technologies start to replace CGI as web application development. There are numbers of technologies which try to dominate web development like ColdFusion, Server-side JavaScript (SSJS), PHP, Servlet, JavaServer Pages (JSP), Active Server Pages (ASP), and Active Server Pages.NET (ASP.NET).

Servlet and JSP pose numbers of benefits. There are some benefits which are described as follows:

- **Performance:** In Servlet there is no process creation for each client request like CGI technology. Servlet container process handles each request. After finishing one request it stays alive inside memory and waiting for another request to process.
- **Portability:** This technology is portable like other java technology. Application can be moved to any operating systems without big problem.
- **Rapid development cycle:** Servlet has a huge access to rich java libraries. It increases the functionality of Servlet technology. Access to rich libraries helps java developer for rapid development and speed up their work.
- **Robustness:** Java Virtual Machine manages Servlets and handles garbage collection or memory leak. This facility makes Servlet technology robust.
• Widespread acceptance: java is very familiar and famous technology around worldwide. Numerous vendors work on java technology. There are lots of components which are available in java technology. It is easy to pick a target component which saves precious time. [11]

Servlets are java classes. It runs on web server called Servlet container. In early days it is named Servlet technology. It works with request response model based on HTTP (Hypertext Transfer Protocol). Servlet container supports HTTP protocol. A client request passes to server through HTTP protocol as well as a server response. For more secure connection Servlet container supports HTTPS protocol which is HTTP over SSL (Secure Sockets Layer). Figure 1 shows architecture of Servlet application

![Figure 2: The Servlet application architecture](image)

Servlet also comes with HTML pages and images. So it is not appropriate for Servlet container to handle static content because static content will be faster if any HTTP server like Apache or IIS web server handles this. For this a revised architecture shows that a web server in front handles all client requests. Web server filters all static contents and sends dynamic content like Servlet requests to Servlet container. Figure 2 shows a complete architecture for Servlet applications.

![Figure 3: The Servlet application architecture with HTTP server](image)
When a Servlet is requested it first comes to Servlet container. Servlet processes the user request and sends back response to Servlet container. Servlet container again sends back the response to the user. Servlet stays in memory as long as no memory shortage in container. Servlet keeps inside container to process another request. Servlet container compares the timestamp of loaded Servlet with the Servlet class file. If class file timestamp more recent then Servlet is uploaded into memory. This feature prevents Servlet be uploaded each time one update the Servlet. Figure 3 shows a working process of a Servlet inside Servlet container.

Figure 4: Flowchart of Java Servlet processing. [11]
3.2. **Java Server Pages (JSP)**

JSP is another java technology. JSP is introduced in the time when Servlet technology was at its pick. JSP is not the competitor or replacement of Servlet technology. JSP is an extension of Servlet technologies. Both technology works together to make the development application easy. Java developers use both technologies in java web applications. It is like a common practice for developers to use both technologies.

Using JSP is easy for developers. Without much knowledge underlying API one can write JSP application. JSP uses similar technique like, work with HTTP request and response.

The Servlet is introduced by Sun in 1996 to use dynamic content in web application. Within very short time it is gained as one of the popular technologies for internet development. However java programmer know it is difficult to write long HTML page that includes little code inside sevlet. Appendix B.5 shows a glimpse of Servlet code.

Here inside doPost method most of the codes are static content. HTML tag is embedded inside string and sent using the println method. When this HTML is long then it is difficult for a programmer to write static code in that fashion. Another problem has arisen when modification of a single HTML code needs java programmer. Java programmers have to work under the guidance of web design. Application layer is not different from business layer. JSP pages add lots of advantage in Servlet technology. It is not the competitor for Servlet technology. It gives more authorization to Servlet technology. Combining fixed and static template data and dynamic data become easier. In JSP page a java programmer can easily edit the page with his code and save it as a .JSP file. Later stage if any changes are required by the web designer, he simply can open the page and edit the static content. It saves lots of money and time for developing java web applications.

The following section defines some advantages JSP technology with other similar technologies.
Dynamic

- JSP vs. Active Server Pages (ASP): first advantage of JSP is, it is portable. Java web applications can easily be moved to other operating systems and non Microsoft Web servers. JSP dynamic part is written in java and java is easier and more powerful to use.

- JSP vs. Pure Servlets: There is no difference between Servlet and JSP. JSP has the similar features that Servlet provides. But it is better and more formal approach to write or edit regular HTML than have many println statements to generate HTML. Different people task has been easier to managing the Servlet applications. Web designer can build the html and send it to Servlet programmer to add dynamic contents or vice versa. Other way when it comes for edit, designer can work indecently without the help of java programmer.

- JSP vs. Server-Side Includes (SSI): SSI is used for including external program inside static web page. On the other hand in JSP, programmers can use Servlets instead of another separate program. SSI does not support for large scale program like form data or make database connections.

- JSP vs. JavaScript: Really there is no comparison between the two technologies because JavaScript is a client side script so it can not access server side script like databases, browse catalogs, details information.

- JSP vs. Static HTML: HTML page only contains static information. It does not contain dynamic contents. [10]

4. Application Design
We have divided the architecture into two parts, first we have described the database architecture and then we have described about the implementation architecture of the application in both technology. We have implemented the same database and implementation architecture in both technologies.
4.1. Database Design
We have designed the same five tables with their attributes in both SQL Server and MySQL databases. We have also shown the dependencies between the tables below in diagram. There is relation between some tables. Category table has foreign key in the Product table to show, add and update the products by specific category. Product table has the foreign key in Product_Order table to get the purchase product information and Registration table also has the foreign key in the Product_Order to get the user information in Product_Order table.

```
<table>
<thead>
<tr>
<th>Category</th>
<th>Registration</th>
<th>Product</th>
<th>Product_Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK</td>
<td>PK</td>
<td>PK</td>
<td>PK</td>
</tr>
<tr>
<td>Category_id</td>
<td>user_id</td>
<td>Product_id</td>
<td>ProductOrder_id</td>
</tr>
<tr>
<td>Category_name</td>
<td>user_name</td>
<td>Product_name</td>
<td>FK1 FK2</td>
</tr>
<tr>
<td>Category_description</td>
<td>user_email</td>
<td>Category_id</td>
<td></td>
</tr>
<tr>
<td></td>
<td>user_pass</td>
<td>Product_quantity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>user_type</td>
<td>Product_picture</td>
<td></td>
</tr>
<tr>
<td></td>
<td>country</td>
<td>Product_description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>city</td>
<td>Product_price</td>
<td></td>
</tr>
<tr>
<td></td>
<td>access_type</td>
<td>Product_status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Admin_id</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Admin_name</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Admin_pass</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 5: Database Diagram

4.2. Implementation Design
We have developed the design and implementation code separate from each other to make the system more upgradable, reliable and secure. We have developed the some code classes that provide the functionality to design classes to fulfill the requirements. Every design class uses the code class in both technologies. We have created the relationship between the code classes that is shown in the diagram.
4.3. User Interface

We have described our application how it works by showing the diagram of some Web pages. When user will open the website then home page will start by default. Home page contains the menu bar, registration, login, search, left panel, right panel and list of all products. User can see the detail of the product by clicking on the any product. User can also search the specific product by giving input query in search textbox. User can also view the specific category products by clicking the any category link in the left panel of the page. User can also view the detail of new product by clicking on new product in right panel. User can also get the registration by clicking the Registration link that is on the right top of the page. Client of the system can also login by click the login link, when user clicks the login link it open the login page. If the user is Admin then the redirect to admin panel and if the user is client then the user redirect to the client panel. Admin and client both have the different rights. Admin can update the system but client can not update or change the system. Client just can use the system. There is difference between...
the visitors and the client of the system. Visitors can just search and view the product but the client can view search and purchase the products.

![Home Page](image_url)

Figure 7: Home Page
When user click Add to Cart button in product detail page, then the User shopping cart page will open. Shopping cart contains the some information of the selected products,
user can also update the quantity of the product. User can delete the product from the cart. User can also purchase the products by clicking the Purchase button but user must be login before clicking this button, if user not login and click this button then the user will redirect to login page automatically.

Figure 9: User Shopping Cart Page
Admin can login by clicking the login link in home page. Admin has different design panel and different menu bar. Admin can add product, update product, add category and update category. In diagram we have shown how admin can add the products in database. When admin click on Add Product link then add product page will open. Admin must select the category of the product from the dropdown list and then give information of the product by typing the input in textboxes and then click on the submit button. If Admin wants to add more products then must click the Add New button.

Figure 10: Admin Add Product Page
Figure 11: .NET application Implementation structure
5. **Implementation Application Steps in .NET c#**

There are several implementation steps in the development phase that we have described from the start to end of the project development phase.

### 5.1. **Install VS 2010 and SQL Server 2008**

First we have installed the Visual Studio 2010 professional version with SQL Server 2008. Before starting the project in ASP.NET 4.0 we have downloads some ASP.NET tutorial to learn and understand the development of code, how it works and what are the main objects in development phase? We learn the basic concepts from the tutorial that we need to use in our project for example, how we can implement the separation of the code in presentation layer and business logic layer? How can we implement the design layer of the project? How can we implement the business logic layer using C# language? How can we communicate the ASP.NET with SQL Server 2008? We follow these tutorials and we developed a simple example according to the tutorial. After practice the example we start to implement our actual project.

### 5.2. **Create Database & Connection**

At first we have read some tutorial about the SQL database 2008 that how it works and how can we create the database with tables. We also lean from the tutorial to create the Store procedure in database. Store Procedure makes the database more secure and reliable. We have described step by step to create the databases that are following.

- Open the web Application in visual studio 2010 and go to server explorer.
- Right click on data connection and select Create new SQL server database.
- Select windows authentication option.
- Give the server name and database name.

We have also described the database tables and store procedure step by step when the database is created, these steps are following.

- Open the database menu and right click on the Tables tab.
- Select the Add new table option.
- Give the Column name and its data type.
- Save the table by giving the name of table.
We have created some tables in database to manage our system. We have also described the purpose of these tables. Registration table is use to save the personal information of the client. Admin table saves the personal information of the Administrator of the system. Product table is use to save the information about products. Category Table is use to save the information about Category of the product.

We have also created some store procedure in our database that are, addcategory store procedure uses for enter category information in Category table, AddProduct store procedure is use for enter the information about product into Product table and ClientReg store procedure is use for enter the information about the client in Registration table. We have written the connection string in WEB.Config that is responsible to have connection with SQL Server for all ASP.NET application. We have described the connection string in Appendix B.5.

5.3. **Create Web Application Project**
- First open the Visual Studio and select the File >> New >> Web.
- Select the C# from the installed template.
- Select the ASP.NET empty web site. If you select the ASP.NET Web site, that contains the simple implemented code example. If you want empty web site then select the ASP.NET empty web site.
- Give the Application name and brows path. And finally web application project is created.

5.4. **Create and Implement Classes**
First we created the classes for business logic layer for the customer panel and admin panel. To create the classes there are some steps that we have followed.

- Go to the Solution Explorer and right click on the web application and then select Add ASP.NET Folder and then select App_Code.
- Right Click on the APP_Code and select the Add New Item.
- Select C# language from the Installed Template on the left corner and then select the Class.
• Give the name to Class and the click on Add button. You will see the new class is created under the App_Code tab.

There are many classes in which we have implemented business logic of our project. We have explained the functionality and purpose of these classes in our report that are below. Client.cs class is responsible for the registration functionality. This class uses the Registration() function. This function get the connection string from the Web.config file and uses the store procedure to save the user input in database. User.cs class is responsible for the user authentication. If the user is client then this class redirect user to the client home page and if user is the admin then this class redirect the user on admin home page. This class has one function called Login(). This function uses the web.config for getting the database connection string. Product.cs class is responsible for the product management. This class contains the several functions to manage the products. Admin uses the AddProduct() and UpdateProduct() functions to add and update products in database. Admin uses the Delete() function to delete the product from the database. User uses the Search() function to search specific product from the database system. GetNewProduct() and getItem() functions are use to show new products and all the products respectively in home page for all users. Product_Category.cs class is use for the category management. This class contains the several functions for the user and admin. Admin can use the AddCategory(), UpdateCategory(), delcategory() functions for the add category, update category and the delete category in the database respectively. GetAllCategory() function use to display all the categories in the main page. Product_Category.cs class contains the MakeOrder() function that is responsible to make the transaction for the client when client want to purchase the product. This function also update the purchased quantity of the products in database that customer buy the products. ShoppingCart.cs class provides the dynamic functionality to the user. This class not interacts with the database. This class is use to manage the shopping cart. This class contains the AddProduct(), UpdateProduct and DeleteProducts functions to manage the user shopping cart. AddProduct(), function is use to add product in shopping cart, if the product is already in shopping cart then this function increase the quantity of that product. UpdateProduct() function is use to update
quantity and price of the products in shopping cart and DeleteProduct() function is use to delete product from the shopping cart.

5.5. **Create & implement Master pages**

In web application this is very important to have consistent layout, design of the web. ASP.NET provides the feature called Master Page that contains the common design and functionality for the group of web pages. Master page provide the content place holder for the other ASP.NET pages. In content pages developer can put the design and functionality in content place holder. When the developer compiles the website, the output is the collection of the master page and content pages. For example if you want to make the same design header and footer then just make the one time in master page and do the functionality of other pages in content placeholder. The output will be the combination of master page and content page.

To create the master there are some steps that are following

- Go to Solution Explorer and right click on the web application.
- Select the Add new Item.
- Select the language C# and the select the Master Page and then give the master page name and path, then click on Add button.

We have created 2 master pages in our web application. One master page contains the Client panel pages called MasterPage.master and other master page contains the Admin panel pages called Admin.master. Master page has the extension .master and content page has the extension .aspx .In MasterPage.master we have design header, footer, left panel and right panel. In header we have created the menu bar, Registration, Login, Main Picture and in left panel we have created All categories and in right panel we have created the new coming products. In other master page Admin.master we have created the header and footer for the admin side pages. Header contains the menu bar and logout link.
5.6. **ASP.NET Content Pages**

To create the ASP.NET content pages there are some steps that are following

- Go to Solution Explorer and right click on the web application.
- Select the Add new Item.
- Select the language C# and the select the Web Form and then give the Web Form name and path, then select the master page check box and click on Add button.

After following these steps asp.aspx and its ASP.aspx.cs file also created. ASP.aspx page is used for ASP.NET controls and HTML designing and ASP.aspx.cs file is used for C# coding for the business logic. There are several ASP.NET pages with their functionality that we have described in our report that are following by their names.

When user will opens the web site Home.aspx page will appear initially. This page contains the Data List control. Data list control shows the all product by default but if the user want to search the specific product and then this data list shows the product related to the search query. Home.aspx.cs uses the Product class to fulfill the business functionality.

Login.aspx contains the labels, textboxes and the Login button. When user will enter the user name and password then click the login button, if the user name input is missing then the label shows the username is missing and if the password input is missing then the label shows password is missing and if the username and password impute are wrong then the label shows the invalid username or password message. If the user is client and username and password input has no mistake then this page will redirect to the client panel, and if the user is Admin and username and password has not mistake then this page will redirect to the Admin panel. Login.aspx.cs uses the User class for the business functionality.

Registration.aspx page contains the labels, textboxes and Registration button. User can get the registration by giving some information in this page. User name must be the user email id, of the email id is already exist then the label shows the message “Email id is already exist” and user need to give the another email id input. Registration.aspx.cs uses the Client class for the business functionality.
Category_Product.aspx contains the List control. Data list control shows the all the category of the products. Category_Product.aspx.cs uses the Product_Category class for the business functionality.

ProductDetail.aspx contains the labels, product picture and Add to Cart button. This page comes when the user click on any product and after click on product redirect to this page with detail information. ProductDetail.aspx.cs uses the Product class for the business functionality. If user click on the “Add to Cart” button, then this button redirect to the UserShoppingCart.aspx with some information.

UserShoppingCart.aspx page contains the Data Grid, Textboxes and buttons. Data Grid contains the selected product information, quantity and Delete product button. User can update the quantity of the product by giving input in quantity textbox and click on Update button. User can delete the Cart product. User can click on the Purchase button. If the user click on the Purchase button then this button must check the user is login or not on this web. If user is not login then this button will redirect to the Login.aspx page. If the User is already login then this button will redirect to the next page “Checkout.aspx” page with some information.

Checkout.aspx page is responsible for the purchasing products, when the user press checkout button then the number of quantity of the product also decrease in the table. This page uses the Product_order class for the functionality.

Addcategory.aspx page is use for admin panel page that admin can add the category. This page contains the textboxes, labels and button. This page uses Product_Category.cs class for the business logic layer.

UpdateCategory.aspx page contains the data grid, textboxes and button. When Admin will click on the update category link in admin panel then this page shows the entire category in data grid. Admin can delete the category, if admin click on the update link then the old information come in textboxes and admin can change the input and click on the Update button.
Admin can also add the product by using the AdminProduct.aspx page. This page contains the drop down list, textboxes, labels and button. This page uses the Product.cs class as the business logic layer.

Admin can also update the product by using the UpdateProduct.aspx. This page uses the dropdown list, data grid, textboxes, labels and button. First admin must select the category to update the product. This page uses the Product.cs class as the business logic layer.

6. Implementation Application Steps in JSP Netbeans

6.1. Install NetBeans IDE 6.8 and MySQL
NetBeans latest IDE can be found in the following link http://netbeans.org/downloads/. One can download target NetBeans IDE for their operating system and install it without any problem. The web application which I developed used NetBeans IDE 6.8. Those who want to work now; they should download the latest version. Without hassle one can install it in C drive. I used MySQL database because I was familiar with MySQL before for my previous experience working on PHP and MySQL. Java DB, a java based open source database is also available for use. Anyone can use Java DB for web applications as well. The official website of MySQL is http://www.mysql.com/. MySQL downloads and related information is available in that particular site. In my project I had preferred to download XAMPP for Windows from apache friends site.

http://www.apachefriends.org/en/xampp-windows.html . XAMPP is an installer package which comes with apache web server, MySQL, PHP and Perl. I used XAMPP installer to use MySQL database. After installing XAMPP, start the MySQL service from XAMPP control panel. There are some MySQL manager and control tools which are very easy to use and handle database functions. These tools come with rich GUI (Graphics User Interface). One of the examples of this kind tool is SQLyog which can be purchased and downloaded from http://www.webyog.com/en/
6.2. **Create Database & Connection**

We have used same list of tables according to database diagrams. MySQL database is very simple to use. There is Query analyzer. We can create tables by writing Queries. Others can use table crate form to create tables. All the data types are available. Binary data can be saved as blob as well datetime can be used to store dates. The following figure demonstrates product table in Query.

```sql
CREATE TABLE `product` (  
    `Product_id` int(11) NOT NULL AUTO_INCREMENT,  
    `Product_name` text NOT NULL,  
    `Category_id` int(11) NOT NULL,  
    `Product_quantity` int(11) NOT NULL,  
    `Product_picture` text,  
    `Product_description` text NOT NULL,  
    `Product_price` double NOT NULL,  
    `Product_status` smallint(6) DEFAULT NULL,  
    PRIMARY KEY (`Product_id`)  
) ENGINE=MyISAM AUTO_INCREMENT=28 DEFAULT CHARSET=latin1
```

Figure 12: Create table Query

At first I open NetBeans IDE 6.8 GUI. The interface of NetBeans is very easy to use and user friendly. Different tools and windows or navigation buttons are available for using many features. At initial step I made a database connection with MySQL database from Services window->Databases. There Java DB database is also available. If anyone wish he can use. One has to right click over Databases tab. Select New Connection, New Database Connection form will open. Select MySQL from Driver Name drop down list. Other required fields have to be filled up. The following Figure 5 shows the screenshot of Database connection window. After creating connection with Database, one has to start the connection. Now it is ready to start MySQL database in the project.
6.3. **Create Web Application Project:**
Open the Projects window. Click over New Project button. Select Java Web->Web Application. Next step is to select Project Name, browse Project Location, and click next in window. Select server from server drop down list. If no server is selected user will create a server from Services->Servers-> right click->Add Server. From our project, we have used GlassFish v3 web server. Java EE 6 Web is to be selected from Java EE version tab. Click Finish. Inside projects tab, web application project files will be created with folders. One has to put all the source classes inside Source Packages folder. Inside Libraries folder library packages are available. All JSP and XHTML files must be kept inside Web Pages folder. Figure 6 shows a screenshot of project folder.
In the above figure we see the list of class files inside Servlets folder. Classes are the same like our class diagram for both technologies.

6.4. Create and Implement Classes

We have used JavaBean or Java class files to implement our business logic. This class files is used from JSP pages using the useBean tag. The following tag illustrates the use of Products class inside JSP page.

\[
\text{<JSP:useBean id="Products" scope="application" class="Servlets.Products"/>}
\]

Before using Products class one has to instantiates the Products JavaBean and puts it in the Servlet context object. This is done inside controller Servlet. Controller Servlet loads with the project. It reads from the deployment descriptor and initializes all the global variables for the whole applications as well. Controller Servlet also processes the entire incoming request inside it doPost method and forwards each request to one of the pages. Controller Servlet initializes all the class files with the starting of project so that classes
can be used frequently from JSP pages. The code which is mentioned inside Appendix B.1 gives an idea about the initialization of class files inside Controller Servlet page.

To create a class file one has to right click over Source Packages-> New-> Servlet. If someone wants to put the file inside a particular folder he will choose a location from location field. Click finish. By checking Add information to deployment descriptor (web.xml), one can insert Servlet related information into descriptor file. It is done, Servlet class file is created. The code in Appendix B.2 shows getCategoryProduct() method from Product_Category class.

getCategoryProduct() method returns list of products for a particular category id from product table. We can see the return type is ArrayList.

Inside Appendix B.3, GetallCategory() returns an array which contains all the category from category table.

Figure 15: Structure of Product_Category class file.

In the above figure we see the list of class files like ProductCategory.java, Product_order.java in left side. At Right side, Product_Category.java file has been opened in main tab. Now we have the idea how to create class files inside NetBeans IDE.
Now I will show how I have called the methods of this class from JSP page. Appendix B.4 gives an example of menu JSP page.

At the beginning of this code we see the `useBean` tag with class path. Inside HTML body under “<% %>” tag we see calling method, `Product_Category.GetAllCategory()` which fetches all the categories from category table by `Product_Category` class. It prints the list of categories with links to `BrowseCatalog` page. We see when user clicks the link it first goes to Controller Servlet then bypasses to `BrowseCatalog.JSP` page. Variable base contains the path of Controller Servlet.

### 6.5. Application Pages in JSP:

Application part we have used JSP pages. Though there are other latest technologies like Java server faces (JSF) or facelets. We prefer to use JSP because it is easy to use and there are lots of tutorials and help for JSP. This technology is flexible to use. We can write code anywhere inside JSP page. It is similar like PHP. Because we had previous experience to use PHP so we choose JSP technology for application part. Basically application part is divided into two parts. One is admin part and another is user. For admin part the following functions Add Product, Update Product, Add Category, Update Category, Manage Users are been implemented with JSP pages. On other hand user can browse catalog, view new product list, search products, add products to shopping cart, make registration, log in, and purchase products. These are the main features for users.

The below section explains some pages of Admin part.

After log in as an admin user redirects to admin index page which is `HomeAdmin.JSP`. Admin index page includes three pages. These are admin menu page on the left hand side, admin header page at top and admin content page. Admin header page contains all the links of Admin panel. The page includes by JSP include tag. The following code includes `HeaderAdmin.JSP` page.

```
""<JSP:include page="HeaderAdmin.JSP" flush="true"/> ""
```

Add product is used to add new products to shopping categories. It is a simple form with product details like product name, product price, quantity, description, product image.
There is also an option to make added product as special one. When new product option is checked in the process of adding product then newly added product is viewed in the right side of shopping site. In the form all the list of categories come from category table. It shows in the select box. Before adding product a particular category must be selected from select box, product price, quantity, description should be filled up. Whole validation of the page is done through JavaScript. The same page is also used to edit existing product. Basically there are two forms in the page. When the page opens from edit link then edit product form open with target data in the field otherwise it opens with add product form.

Update product page shows all the list of products for a particular product category. This shows the details of a product with two links. One is edit link another delete. When admin click on delete link the particular product deletes from database and it is done by Delete Product page. User clicks on edit link and particular product id send to Add Product page. Then Add product page opens with edit form. This is the way update is done for each product.

Add category page also works similar way like add product page. The size of the form is smaller than product form. There are category name and category description.

It is similar like update product page. One difference is all the categories come directly and show in the table.

The following section explains some of the important pages of shopping cart.

Browse catalog page shows all the list of products. When user selects a category and clicks then it goes to catalog page and views lists of products for a particular product. Basic information of a particular product like product name, description and price is showed in side HTML table. Product_Category class is been used here. When a user clicks on details link of a particular product, it redirects to product details page.

Product details page shows details about product with an Add to shopping cart link. It uses product class. It sends product id to the method of Product class. Product class returns an array list with product details. The page shows image of product inside image
HTML tag. There is a link to add a product in shopping cart. When one clicks the link Add to Cart, the product adds to shopping cart and user redirects to shopping cart page.

It is very important page for a shopping site. Shopping cart page uses shopping cart class. We have used Hashtable here. When a user adds a product, we fetch all product details from product table and add the product into Hashtable. After adding the product we put Hashtable variable inside session variable. Shopping cart can be updated. User can delete item from shopping cart. Also one can update product quantity. Session is managed by HTTP Session variable. We have used level, submit button and text box inside form. When someone clicks update button then the quantity is updated both inside page and Hashtable. The product id deletes from Hashtable with clicking Delete button. All these functions are handled by Shopping Cart class file. The page redirects to order page with clicks over Check Out.

Order page checks first that user is logged in or not. If user is already logged in then it opens the order form to finish the form. Order process page uses Product_order class. After successful purchase, it inserts order details and delivery information into database. Inside order form text boxes is being used.

Registration page contains the fields like user email, password, address and mobile. When user clicks submit button then first it checks that user already logged in or not. Registration process is done by the User class. With successful registration it shows success message. Registration field contains text fields, password text field, table and form with submit button.

Both admin and user uses the same login form. The form validation is done by JavaScript. Log In process page uses User class. First it checks through User table, if finds match then returns with a particular value and redirects to User index page. With no match in User table it searched through Admin table and with true return opens Admin Index page.

It is the first page of shopping site. When a guest user browses the site this page opens. It is similar like admin index page. Two different pages are includes inside it. Header page which includes at top comes with all the links and header image. Left menu page
includes at left side which contains all the list of categories. Right side of the page shows newest products inside scrollbar. User can view the newest product and click the link to visit directly to the product details page. Search option is also there with search field. One writes something inside search field and with clicking search button search results are showed. The center of the page contains thumbnail images of all the products. Paging has done in the page. Simple table is used here. The page uses Product, Product_Category classes etc.

7. **Experience of Implementation in Both Platform**

There are lots of design difference in between ASP.NET and JSP. ASP.NET comes with complete set of control tools. There are not enough control tools for JSP in NetBeans IDE. So it is difficult to manage site. Developer can build site easily on ASP.NET. This technology is easy for rapid and fast development. ASP.NET controls are more powerful and requiring less time to develop. JSF (Java Server Face) comes with numbers of control tools like ASP.NET. But as we use JSP in our project, so we find little control tools for JSP. ASP.NET control tools tightly couple with the visual studio development environment with lots of helper wizards. Here developer can easily edit the controls with visual editors. In our project it takes a lot time to code in JSP. But in .NET it was quite easy to code using different helper wizards of control tools. One feature of ASP.NET is Master Page. In JSP we used normal HTML code with JSP include tag. Header and Menu two different pages include inside index page. So there is lots of coding in JSP. ASP.NET provides the big feature master page to implement the consistent and stable design. The developer does not need to implement header, footer or menu in each application design page. Master page handles all. One needs to put different controls and functions inside content place holders of different pages. We include header and menu pages inside all the pages of shopping site for JSP. Another feature of JSP is any changes in codes inside

ASP.NET and JSP both languages are use for the server side scripting. ASP.NET is much slower as compared to JSP if we change any code. There is much difference in code
between JSP and ASP.NET languages. .NET framework provides built-in controls and libraries as well in NetBeans IDE there is not so much built-in help like in .NET. Using the NetBeans IDE most of the code needs to write by developer but in .NET developer just needs to use built-in functions to perform the tasks. .NET provides the built-in data list and data grid containers to show the output on the ASP.NET pages. But meanwhile NetBeans IDE not provides the data list container in JSP like in asp.NET, developer need to perform same task in tables or in other manual way. But Netbeans IDE also introduce the JSF (Java Server Face) technology same like as ASP.NET. JSF provides the data list container and some more built-in controls to make the website faster .NET code syntax much easier to understand as compared to JSP code syntax. JSP take much time to write code as compared to ASP.NET. We have also faced the difficulty to search the help for JSP code as compared to ASP.NET.

We have not face the big difference in both data bases SQL server and MySQL except of syntax difference. Both databases provide the security and reliability to the users and work same probably.

8. Conclusion & Future Work
We have developed the two applications in both technologies Visual Studio 2010 and NetBeans IDE 6.8. We have much better experience in Visual Studio 2010 by using ASP.NET with C# as compared to NetBeans IDE by using JSP with Servlet because there is lots of the documentation informs of code examples and tutorials online to learn .NET framework. We found that ASP.NET provides the strong web development environment to develop the design by using many controls and features, for example master page that is use to make the stable design of all application. We would say that .NET provides many built-in functions and code is also short to do some functionality rather then we have written very much code to do same functionality and almost all code manually in Netbeans IDE. We have also found the advantage to use JSP when we need to update any code then we not need to have again compile the application just we need to refresh, this is because JSP compile the code at runtime while as in ASP.NET, when we update some code then we need to have again compile the whole application.
This work, Evaluation of Two Web Technologies can improve by analysis the both application in security risks perspective and evaluate which technology provides the more security mechanism. This is very important in these days to develop online business application very secure and reliable for the users. This web application functionality can also improve by implement the rest of functionality in both applications like Client management functionality.

9. References

1) The TextBox Control, last checked 2011-01-18,
URL: http://www.w3schools.com/aspnet/aspnet_textbox.asp

2) The .NET Framework, last checked 2011-01-18,

3) .NET Framework Overview, last checked 2011-01-18,
URL: http://www.microsoft.com/net/overview.aspx

4) The GlassFish Community Delivering a Java EE Application Server, last checked 2011-01-18, URL: http://glassfish.java.NET/faq/v2/GlassFishOverview.pdf


6) Visual Studio 2010 Product Highlights, last checked 2011-01-18

7) Visual Studio 2010, .NET 4.0, last checked 2011-01-18,
URL: http://www.authorstream.com/Presentation/amrik-155782-new-features-vs-2010-net-4-0-visual-studio-dot-vb-10-education-ppt-powerpoint/

8) Sun Microsystems, last checked 2011-01-18, URL: http://java.sun.com/

9) Sun Microsystems, Java, last checked 2011-01-18,
URL: http://www.NETbeans.org/index.html

10) A Servlet and JSP tutorial, last checked 2011-01-18
URL: http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/
Appendix A : Visual Studio 2010 Implementation

A.1 Implement user class

```java
public class User {

   public User() {

      //
      // TODO: Add constructor logic here
      //
   }

   public int Login(string username, string password) {
      try {
         string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
         SqlConnection conn = new SqlConnection(str);
         SqlCommand cmd = new SqlCommand("", conn);
         SqlCommand cmd1 = new SqlCommand("", conn);

         conn.Open();
         cmd.Connection = conn;
         cmd1.Connection = conn;
         cmd.CommandType = CommandType.Text;
         cmd1.CommandType = CommandType.Text;
         cmd.CommandText = "select User_email,User_pass from Registration where User_email='" + username + "' and User_pass='" + password + "';";
         cmd1.CommandText = "select Admin_name,Admin_pass from Admin where Admin_name='" + username + "' and Admin_pass='" + password + "';";
      } catch (Exception ex) {
         // handle exception
      }
   }

   // other methods...
}
```


13) Your First Cup: An Introduction to the Java EE Platform, last checked 2011-01-18
```csharp
SqlDataReader rd = cmd.ExecuteReader();
if (rd.Read())
{
    return 1;
}
rd.Close();
SqlDataReader rd1 = cmd1.ExecuteReader();
if (rd1.Read())
{
    rd1.Close();
    return 2;
}
else
{
    return 0;
}
}
catch (Exception ex)
{
    throw ex;
}
}

A.2 Implement client class
public class Client
{
    public Client()
    {
        // TODO: Add constructor logic here
        //
    }

    public int Registration(string name, string mobile, string email, string pass)
    {
        try
        {
            string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
            SqlConnection conn = new SqlConnection(str);

            SqlCommand cmd = new SqlCommand("", conn);
            cmd.CommandType = CommandType.StoredProcedure;
            cmd.CommandText = "ClientReg";
            cmd.Parameters.Add("@User_name", SqlDbType.VarChar).Value = name;
```
cmd.Parameters.Add("@User_pass", SqlDbType.VarChar).Value = pass;
        cmd.Parameters.Add("@User_email", SqlDbType.VarChar).Value = email;
        cmd.Parameters.Add("@User_mob", SqlDbType.VarChar).Value = mobile;

        conn.Open();
        int rows = 0;
        rows = cmd.ExecuteNonQuery();
        conn.Close();
        return rows;

    } catch (Exception ex)
    {
        throw ex;
    }
}

public int checkemail(string email)
{

    string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    SqlConnection conn = new SqlConnection(str);
    SqlCommand cmd = new SqlCommand("", conn);

    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "select User_email from Registration where User_email='" + email + ";"
    conn.Open();
    SqlDataReader rd = cmd.ExecuteReader();
    if (rd.Read())
    {
        return 0;
    }
    else
    {
        return 1;
    }
}

A.3 Implement Product class
public class Product
{
    public Product()
{  
    // TODO: Add constructor logic here  
    //
    public DataSet getItem()
    {
        try
        {
            string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
            SqlConnection conn = new SqlConnection(str);

            SqlCommand cmd = new SqlCommand("", conn);
            conn.Open();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "select Product_id,Product_name,Product_picture,Product_price FROM Product";
            SqlDataAdapter da = new SqlDataAdapter(cmd);

            DataSet ds = new DataSet();
            da.Fill(ds);
            conn.Close();
            return ds;
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }

    public DataSet search(string a1)
    {
        try
        {
            string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
            SqlConnection conn = new SqlConnection(str);

            SqlCommand cmd = new SqlCommand("SELECT Product_id,Product_name,Product_picture,Product_price FROM Product where Product_name LIKE "\%" + a1 + "\%"", conn);
            conn.Open();
            cmd.CommandType = CommandType.Text;
            //cmd.CommandText = "select User_email,User_pass from Registration where User_email='" + username + "' and User_pass='" + password + "'";
            SqlDataAdapter da = new SqlDataAdapter(cmd);

            DataSet ds = new DataSet();
        }
da.Fill(ds);
string a = ds.Tables[0].Rows[0].ItemArray[0].ToString();
conn.Close();
return ds;

} catch (Exception ex)
{
    throw ex;
}

//OleDbConnection con = new OleDbConnection(@constr);
//con.Open();
//OleDbCommand cmdadd = new OleDbCommand("SELECT id,Quantity,PName,price,pic FROM product where PName LIKE '%" + a1 + "%'", con);

//OleDbDataAdapter da = new OleDbDataAdapter(cmdadd);
//DataSet ds = new DataSet();
da.Fill(ds);
//string a = ds.Tables[0].Rows[0].ItemArray[0].ToString();
//con.Close();
//return ds;

public DataSet getselectedItem(int id)
{
    try
    {
        int id1 = Convert.ToInt32(id);
        string str =
            ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);

        SqlCommand cmd = new SqlCommand("", conn);
        conn.Open();

        cmd.CommandType = CommandType.Text;

        cmd.CommandText = "select Product_id,Product_name,Product_description,Product_picture,Product_price,Product_quantity,Product_status FROM Product where Product_id='" + id1 + "';"
        SqlDataAdapter da = new SqlDataAdapter(cmd);

        DataSet ds = new DataSet();
da.Fill(ds);
        conn.Close();
        return ds;
    }
}
catch (Exception ex)
{
    throw ex;
}

public DataSet GetNewProduct()
{
    try
    {
        string str =
            ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);

        SqlCommand cmd = new SqlCommand("", conn);
        conn.Open();

        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select Product_id,Product_name,Product_description,Product_picture,Product_price FROM Product where Product_status='1";
        SqlDataAdapter da = new SqlDataAdapter(cmd);

        DataSet ds = new DataSet();
        da.Fill(ds);

        // string a = ds.Tables[0].Rows[0].ItemArray[1].ToString();
        conn.Close();
        return ds;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

///////////////////////////////////////////////////////////////////////// Admin/////////////////////////////////////////////////////////////////////////

public int addProduct(string pname, string pcapacity, string price, string pdetail, string pic, string Category_id, string status)
{
    try
    {

string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
SqlConnection conn = new SqlConnection(str);
SqlCommand cmd = new SqlCommand("", conn);

cmd.CommandType = CommandType.StoredProcedure;
conn.Open();

int a = cmd.ExecuteNonQuery();
conn.Close();
return a;
}

catch (Exception ex)
{
    throw ex;
}
}

public void delproduct(string id)
{
    try
    {

        string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);
        conn.Open();

        cmd.CommandType = CommandType.Text;

        cmd.CommandText = "delete from Product where Product_id='" + id + 
"';

        cmd.ExecuteNonQuery();
        conn.Close();
    
}
A.4 Implement Product_Category class
public class Product_Category
{
    public Product_Category()
    {
        // Constructor implementation
    }
}
public DataSet GetallCategory()
{
    try
    {
        string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);

        SqlCommand cmd = new SqlCommand("", conn);
        conn.Open();

        cmd.CommandType = CommandType.Text;
        cmd.CommandText = "select Category_id,Category_name,Category_description FROM Category";

        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds);
        string a = ds.Tables[0].Rows[0].ItemArray[1].ToString();
        conn.Close();
        return ds;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

public DataSet getCategoryProduct(int id)
{
    try
    {
        string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);

        SqlCommand cmd = new SqlCommand("", conn);
        conn.Open();

        cmd.CommandType = CommandType.Text;
    }
}
cmd.CommandText = "select Product_id, Product_name, Product_picture, Product_price, Product_quantity, Product_description FROM Product where Category_id='" + id + "';
SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();
da.Fill(ds);
conn.Close();
return ds;
}
catch (Exception ex)
{
throw ex;
}

public int AddCategory(string name, string description)
{
try
{
string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
SqlConnection conn = new SqlConnection(str);
SqlCommand cmd = new SqlCommand("", conn);

cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = "addcategory";

cmd.Parameters.Add("@Category_name", SqlDbType.VarChar).Value = name;

conn.Open();
int a = cmd.ExecuteNonQuery();
conn.Close();
return a;
}
catch (Exception ex)
{
throw ex;
}

public void delcategory(string id)
{
try {
    string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    SqlConnection conn = new SqlConnection(str);

    SqlCommand cmd = new SqlCommand("", conn);
    conn.Open();

    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "delete from Category where Category_id='" + id + 
                      ";

    cmd.ExecuteNonQuery();
    conn.Close();
}

    public DataSet getselectedCategory(int id) {
    try {
        int id1 = Convert.ToInt32(id);
        string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
        SqlConnection conn = new SqlConnection(str);

        SqlCommand cmd = new SqlCommand("", conn);
        conn.Open();

        cmd.CommandType = CommandType.Text;

        cmd.CommandText = "select 
                          Category_id,Category_name,Category_description FROM Category where Category_id='" + id1 + 
                          ";

        SqlDataAdapter da = new SqlDataAdapter(cmd);
        DataSet ds = new DataSet();
        da.Fill(ds);

        conn.Close();
        return ds;
    }
}
public int updatecategory(string id, string name, string de)
{
    string str = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;
    SqlConnection conn = new SqlConnection(str);
    SqlCommand cmd = new SqlCommand("", conn);
    conn.Open();
    cmd.CommandType = CommandType.Text;
    cmd.CommandText = "update Category set Category_name='" + name + ",Category_description='" + de + "' where Category_id='" + id + ";";
    int a = cmd.ExecuteNonQuery();
    conn.Close();
    return a;
}

A.5 Implement Web.Config

<configuration>
    <appSettings>
        <add key="ConnectionString" value="server=localhost;Integrated Security=True;database=ShoppingStore"/>
        <add key="CrystalImageCleaner-AutoStart" value="true"/>  
        <add key="CrystalImageCleaner-Sleep" value="60000"/>
        <add key="CrystalImageCleaner-Age" value="120000"/>
    </appSettings>
    <connectionStrings>
        <add name="ConnectionString" connectionString="Data Source=umair-PC\sqlexpress;Initial Catalog=ShoppingStore;Integrated Security=True" providerName="System.Data.SqlClient"/>
        <add name="ShoppingStoreConnectionString1" connectionString="Data Source=umair-PC\SQLEXPRESSION;Initial Catalog=ShoppingStore;Integrated Security=True" providerName="System.Data.SqlClient"/>
    </connectionStrings>
</configuration>
Appendix B: NetBeans IDE Implementation

B.1 Initialize Product_Category class
// instantiating the Product_Category

    Product_Category productCategory = new Product_Category();

// initialize the Product_Category fields

    productCategory.setDbUrl(config.getInitParameter("dbUrl"));

    productCategory.setDbUserName(config.getInitParameter("dbUserName"));

    productCategory.setDbPassword(config.getInitParameter("dbPassword"));

// put the bean in the Servlet context

// the bean will be accessed from JSP pages

    context.setAttribute("Product_Category", productCategory);

B.2 Implement method getCategoryProduct inside Product_Category class

public ArrayList getCategoryProduct(String categoryId) {

        ArrayList products = new ArrayList();

        try {

            Connection connection = DriverManager.getConnection(dbUrl, dbUserName, dbPassword);

            Statement s = connection.createStatement();

            String sql = "SELECT Product_id, Product_name, Product_description, Product_price, Product_quantity, Product_picture FROM product" +

                           " WHERE Category_id=" + categoryId;

            ResultSet rs = s.executeQuery(sql);

            while (rs.next()) {


Product product = new Product();
product.id = rs.getInt(1);
product.name = rs.getString(2);
product.description = rs.getString(3);
product.price = rs.getDouble(4);
products.add(product);
product.quantity = rs.getInt(5);
product.imageurl = rs.getString(6);
}
rs.close();
s.close();
connection.close();
}
catch (SQLException e) {} 
return products;

B.3 Implement method GetallCategory

public ArrayList GetallCategory() throws ClassNotFoundException {
    ArrayList categories = new ArrayList();
    try {
        Connection connection = DriverManager.getConnection(dbUrl, dbUserName, dbPassword);
        Statement s = connection.createStatement();
        String sql = "SELECT Category_id, Category_name, Category_description FROM category order by Category_id desc ";
        ResultSet rs = s.executeQuery(sql);
        while (rs.next()) {

category category = new category();
category.id = rs.getInt(1);
category.name = rs.getString(2);
category.description = rs.getString(3);
categories.add(category);
}
rs.close();
s.close();
connection.close();
}
catch (SQLException e) {}
return categories;
}

**B.4 Implement menu JSP page**

<%@ page import="java.util.*" %>
<%@ page import="Imp.category" %>

<JSP:useBean id="Product_Category" scope="application" class="Servlets.Product_Category"/>

String base = (String) application.getAttribute("base");
<TABLE CELSPACING="0" CELLPADDING="5" WIDTH="150" BORDER="0">

<tr>
  <td bgcolor="#F6F6F6">
    <font face="Verdana">Search</font>
    <form action="ControllerServlet">
      <input type="hidden" name="action" value="search">
      <input type="text" name="keyword" size="10">
      <input type="submit" value="Go">
    </form>
  </td>
</tr>
<tr>
  <td bgcolor="#F6F6F6" valign="top">
    Categories:
  </td>
</tr>
<tr>
  <td bgcolor="#F6F6F6">
    <strong>
      <%=
        ArrayList categories = Product_Category.GetAllCategory();
        Iterator iterator = categories.iterator();
        while (iterator.hasNext()) {
          category cat = (category) iterator.next();
          out.println("<a href=" + base + "?action=browseCatalog&categoryId=" + cat.id + ">" + cat.name + "</a><br>" + category name + "");
        }
      %>
    </strong>
  </td>
</tr>
</TABLE>
B.5 The structure of Servlet

public class TestServlet extends HttpServlet {

    //Process the HTTP GET request
    public void doGet(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {

        doPost(request, response);

    }

    //Process the HTTP POST request
    public void doPost(HttpServletRequest request,
            HttpServletResponse response)
            throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<HTML>");
        out.println("<HEAD><TITLE>Servlet</TITLE></HEAD>");
        out.println("<BODY BGCOLOR=#123177>");

        //Get parameter names
        Enumeration parameters = request.getParameterNames();
        String param = null;

while (parameters.hasMoreElements()) {
    param = (String) parameters.nextElement();
    out.println(param + ":" + request.getParameter(param) + "<BR>");
}
out.println("</BODY>");
out.println("</HTML>");
out.close();

} //End of doPost method