

PROJECT REPORT

(MCSP-060)

On

BY

Enrollment No. 082936318

Under Guidance of

Submitted to the School of Computer and Information Sciences, IGNOU

In partial fulfillment of the requirements

For the award of the degree

|  |  |
| --- | --- |
| SYSTEM REFERENCE MANUAL | |
|  |  |
| Title Page | 1 |
| Approval Letter | 2 |
| Guide Bio-Data | 3 |
| Certificate of Originality | 7 |
| Index | 8 |
| Synopsis | 9 |
| Project Profile | 55 |
| Introduction | 56 |
| Objective & Scope | 58 |
| Project Category | 59 |
| Tools & Platform Used | 63 |
| System Analysis | 64 |
| Requirement Analysis | 65 |
| Feasibility Study | 66 |
| System Requirement Specification | 77 |
| Software Engineering Model Used | 134 |
| Cost Estimation | 138 |
| Project Scheduling | 143 |
| Gantt Chart | 144 |
| Pert Chart | 145 |
| System Design | 146 |
| Project Planning | 147 |
| Modularization Details | 151 |
| Data Flow Diagram | 154 |
| ER Diagram | 166 |
| Data Base Design | 167 |
| User Interface Design | 176 |
| Coding | 202 |
| Implementation & Maintenance | 277 |
| Testing | 278 |
| Security Implementation | 288 |
| Software Maintenance | 294 |
| Future Scope | 295 |
| Limitation of the Project | 296 |
| Annexure | 297 |
| Bibliography | 298 |



PROJECT PROPOSAL

(MCSP-060)

On

BY

Enrollment No. 082936318

Under Guidance of

Submitted to the School of Computer and Information Sciences, IGNOU

In partial fulfillment of the requirements

For the award of the degree

Index

* Title Page 1
* Proforma for Approval of Project Proposal 2
* Bio-data of the Guide 3
* Index 7
* Introduction 8
* Objective 10
* Project Category 11
* Tools & Platform used 12
* Software Engineering Model used 13
* Problem Definition 15
* Requirement Specification 17
* Project Scheduling 20
* Scope Of The Solution 21
* Data Flow Diagrams 22
* Entity Relationship Diagram 34
* Database Design 35
* Modules Used In Project 44
* Security Implementation 47
* Future Scope 49
* Bibliography 50

INTRODUCTION

The use of computers and internet has been expanding significantly over the years. Presently internet has been used widely as a means of communication, as a potential source of information, as a means of entertainment and many more. Within few years after existence of internet, business men realized the possibilities of using internet as a medium of business. This idea kick started the first online business ventures. Modern technology has been developed to the extent that even shopping made possible over the internet. The process of shopping done over the internet is called online shopping. Both products and services can be purchased by online shopping. Shopping Hub is used for business to business transactions or business to customer transactions with applications of electronic commerce. Online shopping malls are just an electronic catalogue of products. Hence to facilitate easier access to required products, almost all major online malls allow searching the entire catalogue. The first step in the online shopping is to search for the specific product categories using integrated search function.

Once you find the required product, put it in a virtual Shopping Cart and continue shopping. After you have added enough items in shopping cart, check out each of them. You can add or delete the contents in shopping cart. The next step is to login using a username and password. Many online malls first require you to register with them before allowing you to pay for the bought items. Enter the address to which you want the product to be delivered. You might also want to select the payment modes

for payment here. After choosing the mode of payment there will other boxes to fill your billing address or the payment details. Some online malls even ask for your email, phone numbers etc. Then wait for the confirmation of your order, you can also modify the order by adding or removing items. You can also cancel order if needed. Keep the printed copy of the purchased order and confirmations for records. In the event of any disputes, you will need to show these documents for verification

OBJECTIVES

An online business is a commerce, which is run completely or partially with the aid of Internet and Internet accessories. *“Shopping Hub - An Online Store”* plays a pivotal role in the online shopping of the multiple items. It reduces the load of the consumer as consumer had to go to the shop then select the item and then purchase it the main problem of transport is also been handled now consumer at home can do the shopping by this system. This system is user friendly system as well as very safe and secure for the shopping the entire password goes in the encrypted form so that no one can access it in unauthorized manner.

The purpose of the *“Shopping Hub - An Online Store”* is basically of saving the time and money and the consumer should get the valuable thing of its cost. Online shopping provides the vast description of the computer hardware and that to the in deep. It becomes very handy for the consumer to choose an item from a large number of items which is not possible in the single shop.

Project Category

The proposed project falls under Web Application category in which sets of technologies are used. Server side technology for creating dynamic web pages. ASP.NET is only one of a set of Internet Technology (ASP.NET) ASP.NET is a powerful and flexible server-technologies that comprise the .NET Framework. It helps to create faster, more reliable, dynamic web pages with any of the programming languages supported by .NET Framework. ASP.NET allows using a far greater selection of full programming languages and fully utilizing the rich potential of the .NET Framework.

Tools and Platform used

Software Requirement

Platform/Environment used : Microsoft Windows 7 Ultimate

Web Server : MS IIS 7

.NET Frame Work 3.5

IDE : MS Visual Studio.NET 2010

Browser : Google Chrome

Front End : ASP.Net using C#.net

Back End : MS SQL Server 2008 R2

Designing Tools : Adobe Photoshop,

Adobe Flash

Hardware Requirement

* Intel Dual Core Processor
* 2.0 GHz with 2MB of L2 Cache memory
* 2GB RAM with 666MHz of FSB
* Monitor that supports 1024 x 768 resolution
* 40 GB Hard disk
* 20 GB External Backup Device

SOFTWARE ENGINEERING MODEL USED

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as spiral life cycle model, it is a system development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping model and the waterfall model. The spiral model is intended for Large, expensive and complicated projects. The spiral model, also known as the spiral life cycle model, is a systems development method (SDM) used in information technology (IT). This model of developments combines the features of prototyping model and the waterfall model. The spiral model is intended for large, expensive, and complicated projects.

The steps in the spiral model can be generalized as follows:

The new system requirements are defined in as much detail as possible. This usually involves interviewing a numbers of users representing all the external or internal users and other aspect of the existing system.

A preliminary design is created for the new system.

A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final products.

* A second prototype is evolved by a fourfold procedure:
* Evaluating the first prototype in term of its strengths, weaknesses, and risks;
* Defining the requirements of the second prototype;
* Constructing and testing the second prototype
* At the customer’s option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development costs overruns, operating-cost miscalculation, are any other factor that could, in the customer’s judgment; result in a less than satisfactory final product.
* The existing prototype is evaluated in the same manner as was the previous prototype, and, if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
* The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
* The final system is constructed, based on the refined prototype.
* The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing the basis to prevent a large-scale Failures and to minimize down-time.

Applications

* For a typical Shrink-wrap application, the spiral model might mean that you have a rough-cut of user elements (without the polished/pretty graphics) as an operable application, add feature in phases, and, at some point, add the final graphics.
* The spiral model is used most often in large projects. For smaller projects the concepts of agile of software development is becoming a viable alternative. The US military has adopted the spiral model for its future combat systems program.

Advantages

* Estimates (i.e. budget, schedule, etc,) become more realistic as work progresses because important issues are discovered earlier.
* It is more able to cope with the (nearly inevitable) changes that software development generally entails.
* Software engineers (who can get restless with protracted design processes) can get their hands in and starts working on project earlier.

PROBLEM DEFINITION

Today, when computerization is essential in each and every aspect of human life to keep pace with the technological changes in the world, the process and activities carried out in various business organizations also need to be refined and automate. Until now, each process and different activities of the Business organization are carried out manually or managed manually which need to keep records, which cause many errors and bundles of paper work. This process may result wastage of lot of time and money, labor and it may not result accurately.

Due to all these above mentioned and many other reasons this project “*Shopping Hub – An Online Store*” is developed. This project aims to develop a complete fully functional independent system with the intention of improving the productivity of the operations of the concerned organization through speedy and accurate processing and efficient storage and retrieval of data. By using this software system the concerned organization do not need to maintain the records manually and which avoid the bundles of paper work. So the basic motives behind developing this project is to Provide a complete solution to manage showroom and to evaluate the effectiveness and efficiency of various processes needed to run the business smoothly. Maintain a database, which facilitates easy and efficient access to the records and information. To eliminate the errors and bundles of paperwork, involve and parallel to increase the accuracy and consistency of information. In addition, save on time efforts and money.

This system provides various reports, which help the administrator in decision-making regarding products details, product stock and even sales and purchases.

This system also helps to gain knowledge of the position and status of the organization. This software system helps concerned organization to analyze, control, consequently improve, and rectify the process in order to make this concerned a profitable venture.

The Great interest of management of organization in computerization to keep pace with industry is also a main reason of developing this software system.

The major problems routinely experienced in organization during the analysis may be summarized as lack of coordination between various divisions of the organization, lack of information on the status of resources, lack of information scheduling required resources lack of sale, purchase, production and consumption information are major reasons behind developing this software system.

REQUIREMENT SPECIFICATIONS

Requirements engineering is difficult. It’s not just a simple matter of writing down what the customer says he wants. A fundamental problem in business is that requirements are inherently dynamic; they will change over time as our understanding of the problem we are trying to solve changes. The importance of good requirements and the underlying dynamic nature of the process mean that we must be as accurate as possible, and yet be flexible. Ineffective requirements practices are an industry wide problem. A more disciplined approach to requirements development and management is needed in order to improve project success rates. A *requirement* is a necessary attribute in a *system*, a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a *customer or user*. Requirements are important because they provide the basis for all of the development work that follows. Once the requirements are set, developers initiate the other technical work: system design, development, testing, implementation, and operation.

Requirements Types:-

Hardware Requirements:-

Performance requirement:- How will user interact with the system, How many users will be using system at a time, what type of user will interact

Interface requirement:- how will the GUI work

Specialty engineering requirement:- check for the o/s requirement or any additional software needed

Environmental requirement:- Infrastructural need like Power supply, HDD space etc.

Software Requirements:-

Functional requirement:- identify the number of function system would perform

Nonfunctional requirement: - how system will generate error messages when a query fails to run to completion or a legacy system is not responding within the allotted time. Or unauthorized access occurs in our system.

How to proceed:-

Customer Needs and Expectations:- (Requirements Analysis Input)

Business requirements:- Managers need access to timely and accurate data on personnel in order to meet operational needs.

User requirements:- The user needs the capability to search on personnel across the entire company by predefined skill sets.

Product requirements :- Data formats shall be translated across legacy system boundaries into the format supported by the local user’s system.

Environmental requirements :- There shall be no operational impact on any user other than the impact on information retrieval caused by having a larger

System Requirements Specifications*(Requirements Analysis Output)*

High-level (or system-level) requirements:- the system shall maintain cross-references for information types contained in the legacy systems.

Functional requirements:- The local user shall be able to search all legacy systems in a predefined local, regional, or national geographical area for personnel meeting a specified skill set.

Nonfunctional requirements:- The system shall make use of the public switched network (PSN) and not require dedicated lines of communication.

Derived (or design) requirements:- Design constraints The system shall use public key infrastructure (PKI) communications security.

Performance requirements:- The system shall support up to 20 simultaneous users without any noticeable degradation of service. The system shall return all available skill sets to the user within 1 minute of initiating a search.

Interface requirements:- The system shall present a look and feel consistent with each local office’s legacy system.

PROJECT SCHEDULING

GANTT CHART

A grant chart or bar chart is the simplest form of formal project management. The Gantt chart is used almost exclusively for scheduling purpose and therefore controls only the time dimension.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10jan  2013 | 13jan  2013 | 15jan  2013 | 23jan  2013 | 4feb  2013 | 8feb  2013 |
| Start | Information Gathering |  |  |  |  |  |
|  |  | Problem Identification |  |  |  |  |
|  |  |  | Requirement  Analysis |  |  |  |
|  |  |  |  | Risk Analysis |  |  |
|  |  |  |  |  | Cost Analysis |  |
|  |  |  |  |  |  | Prototype  Designing |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15feb  2013 | 19feb  2013 | 25feb  2013 | 5mar  2013 | 10mar  2013 | 14mar  2013 | 15mar  2013 |
| Deployment  &validation |  |  |  |  |  |  |
|  | Modules |  |  |  |  |  |
|  |  | System  Integration |  |  |  |  |
|  |  |  | Testing |  |  |  |
|  |  |  |  | Deployment |  |  |
|  |  |  |  |  | Documentation | Finish |

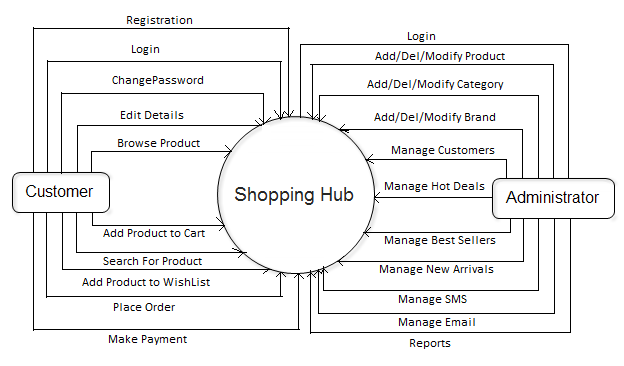
SCOPE OF THE SOLUTION

Modern technology has been developed to the extent that even shopping made possible over the internet. The scope of shopping will remain till the end of this world as without shopping the living is not possible and now a day’s computer is playing a very vital role in each and everybody life. We know shopping is possible through the computer so the scope of this system will increase day by day.

The system “*Shopping Hub- An Online Store*” will serve the computer world and people very much .Computer field is a field where changes occurs in each and every second which the common people are unaware about it so this system brings every product in front of the people which they even don’t know.

DATA FLOW DIAGRAMS

LEVEL 0 (CONTEXT LEVEL)



LEVEL 1 (CUSTOMER)

tbl\_UserAccount

Browse

Or

Search

tbl\_Product

Registration

tbl\_UserAccount

Add

Login

Customer

tbl\_ShoppingCart

Add

tbl\_WishList

Updates

Makes

Payment

tbl\_Order

tbl\_UserProfile

LEVEL 1 (ADMINISTRATOR)

tbl\_HotDeal

tbl\_Brands

tbl\_UserAccount

tbl\_Products

tbl\_Category

tbl\_BestSeller

Administrator

tbl\_NewArrival

tbl\_Stock

tbl\_Admin

tbl\_Suppliers

tbl\_SMS

tbl\_Email

LEVEL 2 (CUSTOMER)

Registartion

tbl\_UserAccount

Details Added

Add Details

User Registered

Enter User Details

Authentication

tbl\_UserAccount

Email &

Password

Verified

Login Succeed

Login Details

Forgot Password

tbl\_UserAccount

Password Reset

Email,

Security Question

& Answer

Password Reset Request

Information

Search/Browse Product

tbl\_Product

Customer

tbl\_Category

tbl\_Brand

Add Product to Cart

Customer

tbl\_ShoppingCart

tbl\_UserAccount

Add Product to Wish List

Customer

tbl\_WishList

tbl\_UserAccount

Place Order & Make Payment

View Selected Item

Add Item

Delete Item

Modify Item

Card Detail

tbl\_Order

Bank Detail

LEVEL 2 (ADMINISTRATOR)

Admin Login

tbl\_Admin\_

UserName

&

Password

Verified

Login Details

Login Succeed

Manage Product/Brand/Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

Manage Hot Deals/Bestseller/New Arrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

Manage Inventory

tbl\_Sales

tbl\_Stock

tbl\_Purchase

Manage SMS/E-Mail

tbl\_SMS

tbl\_Email

Reports Generation

tbl\_Purchase

tbl\_Purchase

tbl\_Purchase

ER DIAGRAM

has

buys

manage

CUSTOMER

PRODUCT

has

added to

makes

made

for

CART

PAYMENT

CATEGORY

1

M

M

M

BRAND

1

ADMINISTRATOR

1

PURCHASE

manage

SALES

maintains

STOCK

M

DATABASE DESIGN

tbl\_Category (PK:Cat\_ID)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Cat\_id | Integer |
| Name | Nvarchar(25) |
| Description | Nvarchar(100) |
| Image | Varbinary |
| Active | Nvarchar(5) |

tbl\_product (PK:Prod\_id FK:Cat\_id Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Prod\_id | Integer |
| Name | Nvarchar(25) |
| Purchase\_price | Integer |
| Sale\_price | Integer |
| Brand\_id | Integer |
| Cat\_id | Integer |
| Description | Nvarchar(100) |
| Image | Varbinary |
| Rating | Integer |
| Active | Nvarchar(5) |

tbl\_brand (PK:Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Brand\_id | Integer |
| Name | Nvarchar(25) |
| Logo | Varbinary |
| Description | Nvarchar(100) |
| Active | Nvarchar(5) |

tbl\_wishlist (PK:Wl\_id FK:Cust\_id,Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Wl\_id | Integer |
| Cust\_id | Integer |
| Name | Nvarchar(25) |
| Prod\_id | Integer |
| Insert\_date | Datetime |

tbl\_Admin (PK:Login\_name)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Login\_name | Nvarchar(30) |
| Password | Nvarchar(30) |

tbl\_Order (PK:Order\_id FK:Cust\_id )

|  |  |
| --- | --- |
| Column Name | Data Type |
| Order\_id | Integer |
| Cust\_id | Integer |
| Order\_date | Datetime |
| Order\_amt | Integer |
| Discount | Integer |
| Shipping\_amt | Integer |
| Tax\_amt | Integer |
| Net\_amt | Integer |
| Shipping\_date | Datetime |
| Shipping\_address | Nvarchar(150) |
| Billing\_address | Nvarchar(150) |
| Status | Nvarchar(15) |

tbl\_ShoppingCart (PK:Cart\_id FK:Cust\_id FK:Product\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Cart\_id | Integer |
| Cust\_id | Integer |
| Session\_id | Integer |
| Product\_id | Integer |
| Quantity | Integer |
| Price | Integer |

tbl\_UserAccount (PK:Uid)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Uid | Integer |
| Username | Nvarchar(25) |
| Password | Nvarchar(25) |
| Hint\_Question | Nvarchar(50) |
| Answer | Nvarchar(30) |
| Status | Nvarchar(5) |

tbl\_UserProfile (PK:Uid)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Uid | Integer |
| Fname | Nvarchar(25) |
| Lname | Nvarchar(25) |
| Dob | Datetime |
| Mobile\_no | Nvarchar(10) |
| Gender | Nvarchar(6) |
| Email | Nvarchar(50) |
| City | Nvarchar(15) |
| State | Nvarchar(15) |
| Country | Nvarchar(15) |
| Aboutme | Nvarchar(100) |
| Pincode | Nvarchar(6) |

tbl\_HotDeals (PK:Hd\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Hd\_id | Integer |
| Prod\_id | Integer |
| Offer\_Price | Integer |
| Description | Nvarchar(100) |
| Active | Nvarchar(5) |
| Image | Varbinary |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_NewArrival (PK:Na\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Na\_id | Integer |
| Prod\_id | Integer |
| Price | Integer |
| Active | Nvarchar(5) |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_BestSeller (PK:Bs\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Bs\_id | Integer |
| Prod\_id | Integer |
| Price | Integer |
| Active | Nvarchar(5) |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_Sms (PK:Sms\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Sms\_id | Integer |
| Type | Nvarchar(10) |
| Mobile\_no | Nvarchar(10) |
| Text | Nvarchar(160) |
| Sendtime | datetime |

tbl\_Email (PK:E\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| E\_id | Integer |
| Type | Nvarchar(10) |
| Email\_id | Nvarchar(50) |
| Subject | Nvarchar(25) |
| Message | Nvarchar(250) |
| Send\_time | datetime |

tbl\_Stock (PK:Stock\_id FK:Prod\_id Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Stock\_id | Integer |
| Prod\_id | Integer |
| Brand\_id | Integer |
| Prod\_name | Nvarchar(25) |
| Description | Nvarchar(150) |
| Quantity | Integer |
| Status | Nvarchar(5) |

tbl\_Supplier (PK:Supp\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Supp\_id | Integer |
| Supp\_name | Nvarchar(25) |
| Address | Nvarchar(60) |
| Prod\_sales | Nvarchar(20) |
| Email\_id | Nvarchar(50) |
| Mobile\_no | Nvarchar(10) |
| TIN\_no | Integer |

tbl\_Sales

|  |  |
| --- | --- |
| Column Name | Data Type |
| Sales\_id | Integer |
| Prod\_id | Integer |
| Cust\_id | Integer |
| Date\_sold | Datetime |
| Qty\_sold | Integer |

tbl\_Purchase (PK:Pur\_id FK:Supp\_id Prod\_id)

|  |  |
| --- | --- |
| Category | Datatype |
| Pur\_id | Integer |
| Supp\_id | Integer |
| Prod\_id | Integer |
| Quantity | Integer |
| Date | datetime |

MODULES USED IN PROJECT

The software will be designed according to the various modules required to fulfill all the requirements uncovered in our requirement analysis. The whole system can be divided into a number of modules. The following modules can be recognized in our project:

There are 7 modules in the project. They are as follows

1.LOGIN MODULE

Login module is the first and the most important module of the project “Shopping Hub – An Online Store”. It plays very important and vital role in the security. As only the authorized user can access it and can do the shopping according to it no other user can access it as it required a user name and password which is used to access the software.

2.ADMINISTRATOR MODULE

This module is used for administrative properties like add, delete, modify etc... Any changes to be made in the product list are done by the help of this module. This module is used by the administrator no one has right to access this module.

3.CUSTOMER MODULE

This module is used for the registration of the customer which holds each and every detail of the customer so that if any fraud case is done by the customer, customer can be easily traced. It keeps the details like name, address, phone number, e-mail address etc.

4.SEARCH MODULE

The very important module of the software as the basis of the software is searching, the product which the customer demands is been searched by the help of this module. This module displays all the items which are presently present for the sale to the customer.

5.ORDER MODULE

This module is used efficiently by the customer as well as by the administrator as customer places the order by the help of this module only and administrator checks the order to be delivered by the help of this module only.

6.PAYMENT MODULE

This module has its own role in the software it is used when the customer had selected the item which he/she has to purchase. This module access the credit card sections for the online payment of the purchase which in turn generate the payment slip which can be printed through the printer very easily.

7.CART MODULE

This is a crucial module which adds various items to the cart then it asks for a confirmation of added items, here the customer can add or delete the items according to his needs and then he is asked for the payment.

SECURITY IMPLEMENTATION

Computer security is an important topic. As e-commerce blossoms, and the Internet works its way into every nook and cranny of our lives, security and privacy come to play an essential role. Computer security is moving beyond the realm of the technical elite, and is beginning to have a real impact on our everyday lives.

It is no big surprise, then, that security seems to be popping up everywhere, from headline news to TV talk shows. Because the general public doesn't know very much about security, a majority of the words devoted to computer security cover basic technology issues such as what firewalls are, what cryptography is, or which antivirus product is best. Much of the rest of computer security coverage centres around the "hot topic of the day," usually involving an out-of-control virus or a malicious attack. Historically, the popular press pays much attention to viruses and denial-of-service attacks: Many people remember hearing about the Anna Kournikova worm, the "Love Bug," or the Melissa virus ad nauseam. These topics are important, to be sure. Nonetheless, the media generally manages not to get to the heart of the matter when reporting these subjects. Behind every computer security problem and malicious attack lies a common enemy—bad software.

## ***Technical Trends Affecting Software Security***

Complex systems, by their very nature, introduce multiple risks. And almost all systems that involve software are complex. One risk is that malicious functionality can be added to a system (either during creation or afterward) that extends it past its primary, intended design. As an unfortunate side effect, inherent complexity lets malicious and flawed subsystems remain invisible to unsuspecting users until it is too late. This is one of the root causes of the malicious code problem. Another risk more relevant to our purposes is that the complexity of a system makes it hard to understand, hard to analyze, and hard to secure. Security is difficult to get right even in simple systems; complex systems serve only to make security harder. Security risks can remain hidden in the jungle of complexity, not coming to light until it is too late.

Extensible systems, including computers, are particularly susceptible to complexity-driven hidden risk and malicious functionality problems. When extending a system is as easy as writing and installing a program, the risk of intentional introduction of malicious behaviour increases drastically—as does the risk of introducing unintentional vulnerabilities. Any computing system is susceptible to hidden risk. Rogue programmers can modify systems software that is initially installed on the machine. Unwitting programmers may introduce a security vulnerability when adding important features to a network-based application. Users may incorrectly install a program that introduces unacceptable risk or, worse yet, accidentally propagate a virus by installing new programs or software updates. In a multiuser system, a hostile user may install a Trojan horse to collect other users' passwords. These attack classes have been well-known since the dawn of computing, so why is software security a bigger problem now than in the past? We believe that a small number of trends have a large amount of influence on the software security problem.

FUTURE SCOPE

The Project “*Shopping Hub – An Online Store*” is generalized software and can be easily used in any online shopping website with little or no change. The changes in software can be easily accommodated. The addition and deletion of the modules in software can be easily adjusted. The software can be enhanced up to any legal extent depending upon user’s requirement. It will be able to serve the organization even if it increases its services. I have completed the project successfully and according to the developed project fully satisfied their needs except certain things because everything is not perfect in this materialistic world and due to certain factors, which directly or indirectly affect the project. We hope that this project will serve its purpose for which it is developed there by underlining success of our project.

Up till now all the activities of the organization are performed manually, it requires more human efforts to manage the each and every aspects of organization which is very costly process. To overcome all kind of problems being faced by organization our project “online-Shopping Computer Hardware” will provide efficient and cost effective solution and always have possibilities of enhancement up any legal extent to satisfy user requirement. This system provides a greater solution for those who are less interacted with the computer system because it provides the graphical user interface facility. This software will help to shopping with wasting your valuable time in crowded malls and large traffic jams.

Bibliography

Websites

|  |  |
| --- | --- |
| URL’s | .NET Resource Description |
| [www.gotdotnet.com](http://www.gotdotnet.com) | All about asp.net |
| [www.123aspx.com](http://www.123aspx.com) | Directory of .NET articles |
| [www.vbws.com](http://www.vbws.com) | Building XML web services with VB. Net |
| [www.soapwebservices.com](http://www.soapwebservices.com) | SOAP Web Services |
| [www.w3.org](http://www.w3.org) | HTML & XML Specification |
| [www.xmlws101.com](http://www.xmlws101.com) | XML web services for ASP.NET |
| [www.mindtools.com](http://www.mindtools.com) | For SAD & Software Engg tools |
| [www.xmlwebservices.com](http://www.xmlwebservices.com) <http://msdn.microsoft.com/> | XML Web Services |

Books

|  |  |
| --- | --- |
| Name of Book | Publication |
| Developing web applications with Vb.NET and ASP.NET | Wiley |
| .NET E-Commerce programming | Sybex |
| Mastering ASP.NET | BPB |
| Complete Rereference ASP.NET | TMH |
| Visual Studio.NET programming | Wiley |
| Visual Basic.Net Programming Black Book | Wiley Dreamtech |

INTRODUCTION

The use of computers and internet has been expanding significantly over the years. Presently internet has been used widely as a means of communication, as a potential source of information, as a means of entertainment and many more. Within few years after existence of internet, business men realized the possibilities of using internet as a medium of business. This idea kick started the first online business ventures. Modern technology has been developed to the extent that even shopping made possible over the internet. The process of shopping done over the internet is called online shopping. Both products and services can be purchased by online shopping. Shopping Hub is used for business to business transactions or business to customer transactions with applications of electronic commerce. Online shopping malls are just an electronic catalogue of products. Hence to facilitate easier access to required products, almost all major online malls allow searching the entire catalogue. The first step in the online shopping is to search for the specific product categories using integrated search function.

Once you find the required product, put it in a virtual Shopping Cart and continue shopping. After you have added enough items in shopping cart, check out each of them. You can add or delete the contents in shopping cart. The next step is to login using a username and password. Many online malls first require you to register with them before allowing you to pay for the bought items. Enter the address to which you want the product to be delivered. You might also want to select the payment modes

for payment here. After choosing the mode of payment there will other boxes to fill your billing address or the payment details. Some online malls even ask for your email, phone numbers etc. Then wait for the confirmation of your order, you can also modify the order by adding or removing items. You can also cancel order if needed. Keep the printed copy of the purchased order and confirmations for records. In the event of any disputes, you will need to show these documents for verification

OBJECTIVE & SCOPE

An online business is a commerce, which is run completely or partially with the aid of Internet and Internet accessories. *“Shopping Hub - An Online Store”* plays a pivotal role in the online shopping of the multiple items. It reduces the load of the consumer as consumer had to go to the shop then select the item and then purchase it the main problem of transport is also been handled now consumer at home can do the shopping by this system. This system is user friendly system as well as very safe and secure for the shopping the entire password goes in the encrypted form so that no one can access it in unauthorized manner.The purpose of the *“Shopping Hub - An Online Store”* is basically of saving the time and money and the consumer should get the valuable thing of its cost. Online shopping provides the vast description of the computer hardware and that to the in deep. It becomes very handy for the consumer to choose an item from a large number of items which is not possible in the single shop. Modern technology has been developed to the extent that even shopping made possible over the internet. The scope of shopping will remain till the end of this world as without shopping the living is not possible and now a day’s computer is playing a very vital role in each and everybody life. We know shopping is possible through the computer so the scope of this system will increase day by day.

The system “*Shopping Hub- An Online Store*” will serve the computer world and people very much .Computer field is a field where changes occurs in each and every second which the common people are unaware about it so this system brings every product in front of the people which they even don’t know.

Project Category

The proposed project falls under Web Application category in which sets of technologies are used. Server side technology for creating dynamic web pages. ASP.NET is only one of a set of Internet Technology (ASP.NET) ASP.NET is a powerful and flexible server-technologies that comprise the .NET Framework. It helps to create faster, more reliable, dynamic web pages with any of the programming languages supported by .NET Framework. ASP.NET allows using a far greater selection of full programming languages and fully utilizing the rich potential of the .NET Framework.

Today, software takes on a dual role. It is a product and, at the same time, the vehicle for delivering a product. As a product, it delivers the computing potential embodied by computer hardware or, more broadly, a network of computers that are accessible by local hardware... As the vehicle used to deliver the product, software acts as the basis for the control of the computer (operating systems), the communication of information (networks), and the creation and control of other programs (software tools and environments).

Software delivers the most important product of our time information.

Software is a logical rather than a physical system element and posses following characteristics:-

1. Software is developed or engineered; it is not manufactured in the classical sense.

2. Software doesn't "wear out."

3. Although the industry is moving toward component-based assembly, most software continues to be custom built.

Our application lies in the category of Web-based software. It consists of web pages retrieved by a browser are software that incorporates executable instructions written in HTML,, C#, Java, Perl etc and data of type hypertext and a variety of visual and audio formats. In essence, the network becomes a massive computer providing an almost unlimited software resource that can be accessed by anyone round the globe.

These general characteristics apply to all Web Applications :-

• Informational. Read-only content is provided with simple navigation and links.

• Download. A user downloads information from the appropriate server.

• Customizable. The user customizes content to specific needs.

• Interaction. Communication among a community of users occurs via chat room, bulletin boards, or instant messaging.

• User input. Forms-based input is the primary mechanism for communicating need.

• Transaction oriented. The user makes a request (e.g., places an order) that is fulfilled by the Web Application.

• Service oriented. The application provides a service to the user (e.g., assists the user in determining a mortgage payment).

• Portal. The application channels the user to other Web content or services outside the domain of the portal application.

• Database access. The user queries a large database and extracts information.

Reasons to use Web Engineering:

Network intensive:- By its nature, a Web Application is network intensive. It resides on a network and must serve the needs of a diverse community of clients.

Content driven:- In our case, the primary function of a Web Application is to use hypermedia to present text, graphics, audio, and video content to the end-user.

Continuous evolution. Unlike conventional application software that evolves over a series of planned, chronologically spaced releases, Web applications evolve continuously and require updating.

Immediacy. Web-based applications have an immediacy that is not found in any other type of software. That is, the time to market for a complete Web site can be a matter of a few days to few weeks to few months. Developers must use methods for planning, analysis, design, implementation, and testing that have been adapted to the compressed time schedules required for Web Application development.

Security. Because Web Applications are available via network access, it is difficult, if not impossible, to limit the population of end-users who may access the application. In order to protect sensitive content and provide secure modes of data transmission, strong security measures must be implemented throughout the infrastructure that supports a Web Application and within the application itself.

Aesthetics. An undeniable part of the appeal of a Web Application is its look and feel. When an application has been designed to market or sell products or ideas, aesthetics may have as much to do with success as technical design.

Internet Standards:- Since the application is used by the wide variety of end user possessing different infrastructural hardware and software. So it becomes difficult to maintain uniformity. Hence some internet standards are given by W3C and IEEE. The web application should follow these standards.

Tools and Platform used

Software Requirement

Platform/Environment used : Microsoft Windows 7 Ultimate

Web Server : MS IIS 7

.NET Frame Work 3.5

IDE : MS Visual Studio.NET 2010

Browser : Google Chrome

Front End : ASP.Net using C#.net

Back End : MS SQL Server 2008 R2

Designing Tools : Adobe Photoshop,

Adobe Flash

Hardware Requirement

* Intel Dual Core Processor
* 2.0 GHz with 2MB of L2 Cache memory
* 2GB RAM with 666MHz of FSB
* Monitor that supports 1024 x 768 resolution
* 40 GB Hard disk
* 20 GB External Backup Device

Requirement Analysis

Requirements engineering is difficult. It’s not just a simple matter of writing down what the customer says he wants. A fundamental problem in business is that requirements are inherently dynamic; they will change over time as our understanding of the problem we are trying to solve changes. The importance of good requirements and the underlying dynamic nature of the process mean that we must be as accurate as possible, and yet be flexible. Ineffective requirements practices are an industry wide problem. A more disciplined approach to requirements development and management is needed in order to improve project success rates. A *requirement* is a necessary attribute in a *system*, a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a *customer or user*. Requirements are important because they provide the basis for all of the development work that follows. Once the requirements are set, developers initiate the other technical work: system design, development, testing, implementation, and operation.

Requirements Types:-

Hardware Requirements:-

Performance requirement:- How will user interact with the system, How many users will be using system at a time, what type of user will interact

Interface requirement:- how will the GUI work

Specialty engineering requirement:- check for the o/s requirement or any additional software needed

Environmental requirement:- Infrastructural need like Power supply, HDD space etc

Software Requirements:-

Functional requirement:- identify the number of function system would perform

Nonfunctional requirement: - how system will generate error messages when a query fails to run to completion or a legacy system is not responding within the allotted time. Or unauthorized access occurs in our system

How to proceed:-

* Customer Needs and Expectations:- (Requirements Analysis Input)

Business requirements:- Managers need access to timely and accurate data on personnel in order to meet operational needs.

User requirements:- The user needs the capability to search on personnel across the entire company by predefined skill sets.

Product requirements :- Data formats shall be translated across legacy system boundaries into the format supported by the local user’s system.

Environmental requirements :- There shall be no operational impact on any user other than the impact on information retrieval caused by having a larger

Population of employees from which to select.

* System Requirements Specifications :- *(Requirements Analysis Output)*

High-level (or system-level) requirements:- the system shall maintain cross-references for information types contained in the legacy systems.

Functional requirements:- The local user shall be able to search all legacy systems in a predefined local, regional, or national geographical area for

Personnel meeting a specified skill set.

Nonfunctional requirements:- The system shall make use of the public switched network (PSN) and not require dedicated lines of communication.

Derived (or design) requirements:- and design constraints The system shall use public key infrastructure (PKI) communications security.

Performance requirements:- The system shall support up to 20 simultaneous users without any noticeable degradation of service. The system shall return all available skill sets to the user within 1 minute of initiating a search.

Interface requirements:- The system shall present a look and feel consistent with each local office’s legacy system.

**Feasibility Study**

This is the first phase, consists of a brief survey of the areas involved, and will result in taking the project into the next phase, postponing development for a period or recommending that no further action be taken. Sometimes it is subdivided into a preliminary investigation (initial study) followed by a more detailed feasibility study. The phase is initiated by management, who perceive the need because of changes or expected changes in the business environment, limitations or failure of existing systems, or the awareness of technological advances relating to the particular are involved in particular systems which competitors are developing.

Information systems projects originate from many reasons: to achieve greater speed in processing data, better accuracy and improved consistency, faster information retrieval, integration of business areas, reduced cost and better security. The sources also vary project proposals originate with department managers, senior executives and systems analysis. Sometimes the real origin is an outside source, such as a government agency, which stipulates systems requirements the organization must meet. When the request is made, the first systems activity, the preliminary investigation, begins. The activity has three parts: request clarification, feasibility study and request approval.

**Request Clarification**

Many requests from employees and users in organizations are not clearly stated. Therefore, before any systems investigation can be considered, the project request must be examined to determine precisely what the originator wants. A simple telephone call may suffice if the requester has a clear idea but does not know how to state it. On the other hand, the requester may merely be asking for help without knowing what is wrong or why there is a problem. Problem clarification in this case is much more difficult. In either case, before any further steps can be taken, the project requests must be clearly states.

This phase (initial study) involves estimating whether or not a development project is worthwhile. Problems with the current automated or manual system are identified, as well as the benefits and costs of an alternative system. If the benefits seem to outweigh the costs (especially when compared with competing projects), a green signal may be given to continue the project, and detailed plans and schedules are drafted for making the system a reality.

The proposed solution to the user’s problem may involve something between dramatic change (completely new system) and slight change to the present system. If the present system is manual and a computer system is proposed, the development project will probably be very large. At the other extreme are small development project that represent slight changes to existing systems, such as sorting information in a different way or inserting subtotals or adding new columns to a report.

**The objectives of this phase are:**

* To determine the feasibility of computerization of a particular system or area of operation.
* To define clearly the objectives, scope and limitations of the project.
* To establish a good working relationship between the user department and the data processing (DP) department.
* To acquaint user management with the approach and method of work in systems development.
* To estimate the resources required for system development, live running and maintenance.
* To identify the likely benefits, which should accrue from the introduction of the system.

During this phase, which should be as short as possible, the systems analyst will be concerned with:

1. How the present system works?

2. The staffing levels involved, their grades and duties.

3. The volume of work: statistics on the various types of transaction, level of overtime working, employment of casual staff, etc.

4. Any current backlog of work and any seasonal influences on the workload.

5. The time taken to process data through the system, delays in issuing management reports, etc.

6. Lists of all documents, files and reports associated with the system.

7. Interfaces with other systems.

An initial study calls for learning as much has possible about the proposed project. At this early point in the project, rough as mates are made of the following:

1. Desired outputs of the system.

2. Available inputs for the system.

3. Time required to complete the changes.

4. Feasibility of such changes.

During the preparation of the initial study, information must be gathered from many sources. Within the organization, information is gathered from users, managers, and system developers. Users and managers must be interviewed to find pout what they need from the system. System developers have a wealth of knowledge of the organizations current capabilities and operations in terms or providing inputs to the system. External information is also critical for determining what new possibilities are offered by vendors in terms of systems available and programs for providing specific types of output. New technologies and developments may allow faster processing, more storage, and lower costs than ever before.

At this point in the project, it is usually very difficult and expensive to specify system characteristics accurately. Often it is impossible to be precise without actually doing much of the proposed work -- including extensive interviewing of users and very detailed studying of procedures. A quick guess about the system’s characteristics is all that is needed at this point.

To develop of rough plan of the outputs of the proposed system, brief interviews are held with the intended users. These interviews result in short descriptions of how the outputs will be displayed or printed on video displays or printed in reports. The inputs required to produce the required outputs must be listed and the sources of these inputs determined. It is important to discover if new files will have to be created or sources of information developed, or if the inputs are already gathered and stored in the current system.

A tentative, general schedule for developing the system should be described. It is determined by the users and the developers and must take into account the urgency of the business need and the limited scheduling resources of the developers.

**Feasibility Study:**

The data collection that occurs during preliminary investigations examines system feasibility, the likelihood that the system will be beneficial to the organization. Four tests of feasibility are studies: technical, operational, political, and economic. All are equally important.

**1.** **Technical Feasibility:** It involves determining whether or not a system can actually be constructers to solve the problem at hand. Some users expect too much of computers, assuming that computers can accurately predict the future, immediately reflect all information in an organization, easily understand speech, or figure out how to handle difficulty problems. Such systems, even if they exist, are not yet available for widespread use.

The technical issues raised during the feasibility stage of the investigation are:

1. Does the necessary technology exist (can it be acquired) to do what is suggested?

2. Does the proposed equipment have the technical capacity to hold the data required to use the new system?

3. Will the proposed system and components provide adequate responses to inquires, regardless of the number or location of users?

4. Can the system be expanded, if developed?

5. Are there technical guarantees of accuracy, reliability, ease of access and data security?

For example, if the proposal includes a printer that prints at the rate of 2,000 lines per minute, a brief search shows that this is technically feasible. Whether it should be included in the configuration because of its cost is an economic decision. On the other hand, if a user is requesting audio input to write, read, and change stored data, the proposal may not be technically feasible.

**2.** **Operational Feasibility:** Proposed projects are of course beneficial only if they can be turned into information systems that will meet the organization’s operation requirements. Simply stated, this test of feasibility asks if the system will work when developed and installed. Are there major barriers to implementation? Here are questions that will help test the operational feasibility of a project:

1. Is there sufficient support for the project from the management? From users? If the current system is well liked and used to the extent that persons will not see reasons for a change, there may be resistance.

2. Are current business methods acceptable to the user? If they are not, user may welcome a change that will bring about a more operational and useful system.

3. Have the users been involved in the planning and development of the project? Early involvements reduce the chances of resistance to the system and change in general, and increase the likelihood of successful projects.

4. Will the proposed system cause harm? The following questions are related to this issue:

Will the system produce result in any respect or area?

Will loss of control result in any area?

Will accessibility of information be lost?

Will individual performance be poorer after implementation than before?

Will customers be affected in an undesirable way?

Will it slow performance in any areas?

Operational feasibility is a measure of how people area able to work with the system. For example, a system may require managers to write BASIC, COBOL, or FORTRAN programs to access data. However, managers probably receive the greatest help from a system when they can concentrate on the problems to solve rather than on how programs should be constructed to solve them.

Issues that are relatively small and seem just minor irritants in the beginning have ways of growing into major problems after implementation. Therefore, all operational aspects must be considered carefully.

**3.** **Political Feasibility:** It is perhaps the most powerful type of feasibility, but it is never formally included in a system study document. This involves the extent of managerial support for changes to the present system. If the boss wants a new computer system, you can bet that it will be installed even if its is not technically, operationally, or economically feasible. Conversely, if the top executive does not want to install a new system, the most convincing reasons to have one will be ignored.

**4.** **Economic Feasibility:** It involves estimating benefits and costs. These benefits and costs may be tangible or intangible. Because of confusion between the types of costs, it is sometimes very difficult to decide if the benefits outweigh the costs.

Tangible benefits may include decreasing salary costs (by automating manual procedures), preventing costly but frequent errors, sending bills earlier in the month, and increasing control over inventory levels. Such benefits may be directly estimated in rupees without much trouble. Intangible benefits may include increasing quality of goods produced, upgrading or creating new customer services, reducing repetitive or monotonous work for employees, and developing a better understanding of the market. Such benefits may be much more important than tangible benefits, but they may be ignored because estimating their rupee values involves pure guesswork.

Tangible costs are easily estimated. They include the one-time cost of developing the system and the continuous costs of operating the system. Examples of development costs are the salaries of programmers and analysts, the prices of the computer equipment, and the expenses connected wit user training., Operating costs include the salaries of computer operators and the costs of computer time and computer supplies. Intangible costs are usually not discussed because they are rarely large. Examples of such costs include those associated with early user dissatisfaction and with the problems of converting to the new system.

A system that can be developed technically and will be used if installed must still be a good investment. That is, financial benefits must equal or exceed the financial costs. The economic and financial questions raised by analysts during the preliminary investigation seek estimates of:

1. The cost to conduct a full systems investigation.

2. The cost of hardware and software for the class of application being considered.

3. The benefits in the form of reduced costs or fewer costly errors.

4. The cost if nothing changes (the system is not developed).

Cost and benefit estimates on each project provide a basis for determining which projects are most worthy of consideration. Each estimate can be analyzed to determine how rapidly costs are recovered by benefits, to calculate both the absolute and interest-adjusted amounts of excess benefits, and to establish the ratio of benefits to costs. All of these factors are considered when developing an overall sense of the project’s economic feasibility.

To be judged feasible, a project proposal must pass all these tests. Otherwise, it is not a feasible project. For example, a personnel record system that is financially feasible and operational attractive, is not feasible if the necessary technology does not exist. Or a medical system which can be developed at reasonable cost but which nurses will avoid using, cannot be judged operationally feasible.

**System Study Report:**

The output from this phase is a formal report called a Feasibility Report or System Study Report. This report may contain:

1. An introduction, which puts the report in perspective and perhaps quotes the terms of reference.

2. The objectives for the system.

3. A commentary on how the present system achieves or fails to achieve those objectives.

4. Interfaces with other systems and the implications of these interfaces.

5. Present operating costs and conditions including organization structure and staffing levels.

6. A first estimate of system development time and a related draft implementation timetable.

7. A schedule of the hardware and other requirements, e.g., availability of system at weekends.

8. Expected benefits, which will accrue from the use of a new system.

9. A cost-benefit projection

10. Conclusions and recommendations.

The study report also covers the following four areas for management:

1. Recommendations for changing the present system: After analyzing all the process and cons of each project proposal, recommendations for changes are made.

2. Effects on operations: If it is recommended to develop or change a system, the report must describe how the system or the changes will affect current operations.

3. Effects on personnel: The effects of the new or changes system on people’s jobs should be carefully described. This includes everything from possible resistance to the new technology to differences in how employees enjoy their work.

4. Details of the changes: All of the detailed information from interviews, preliminary cost and benefit estimations, and other analyses should be included.

The report would normally be supported by working papers and flow charts: It is then up to senior management to decide whether he system should advance to the next stage of the systems development life cycle. Request Approval not all requested projects are desirable or feasible. In fact, some organizations receive so many project requests from employees that only a few of them can be pursued. However, those that are both feasible and desirable should be put into a schedule. Inn some cases, development can start immediately. Butt in most cases; systems staff members are busy on other on going projects. When this happens management decides which projects are most important and schedules them. Many business organizations develop information system plans as carefully as they plan for new products, new manufacturing programs, or plant expansion. After a project request is approved, its cost,, priority, completion

time, and personnel requirement are estimated as used to determine where to add it on any existing project list. Inexpensive projects are not necessarily approved automatically, since many projects compete for scarce resource. Some organizations have long waiting lists (backlogs) of projects. It may take a year or two for the information systems department to begin work on projects submitted and approved today. Later on, when other preceding projects are completed, the proposed application development can be launched. At this time, the collection of data and determination of requirement begin.

**Software requirement specifications (SRS)**

About ASP.NET

ASP.NET is a part of the Microsoft .NET strategy for Web development.

It is a Web development platform that provides the services necessary for developers

to build enterprise-class Web applications. It applies an object-oriented approach to

dynamic Web applications.ASP.NET Web applications are an optimal solution for

managing large-scale business applications. ASP.NET Web applications can be

deployed on the Internet or on an intranet.

Advantage of ASP.NET

* A new development interface under Visual Studio.NET
* Separation of code and presentation logic for better understanding and maintenance.
* Extensive language support like Visual Basic .NET, C#, Jscript and ADO.NET.
* Greater scalability.
* Extensive security support.
* Efficient state management.
* Improved data access using ADO.NET.
* Optimizes the performances of request processing by providing extensive caching support. Easy deployment of applications

Web Application

1. Code is Compiled.

\_\_\_\_\_\_\_

\_\_\_\_\_\_\_

\_\_\_\_\_\_\_

\_\_\_\_\_\_

Code

Compile

Microsoft

Intermediate Language

( MSIL or IL )

JIT compiler

Machine language

Compiles each portion of IL separately

Cache

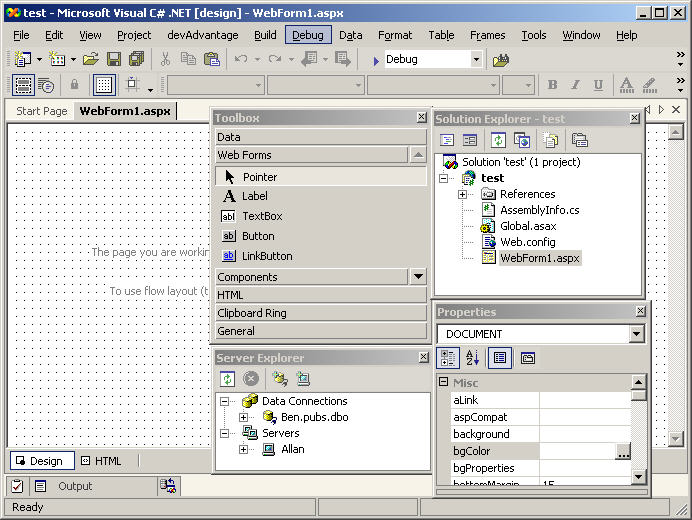
Full Page

Page-level caching – cache a complete page

Cache

Fragment caching – cache part of a page

Part of a Page



Menu bar

And

Toolbar

Toolbox

Solution Explorer

Server

Explorer

Properties

Window

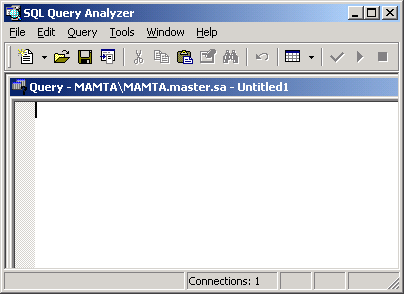
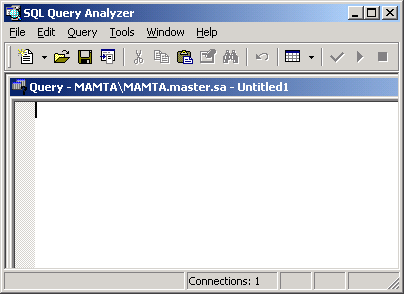
About SQL 2008:-

* SQL Server 2008 offers two tools to work with:
  + - Enterprise Manager
    - Query Analyzer
* The Query Analyzer is a graphical user interface using which developers and administrators can perform everyday tasks such as querying tables, manipulating data within the tables very easily.
* SQL Server Query Analyzer can be launched by selecting the Query Analyzer icon in the Microsoft SQL Server program group.

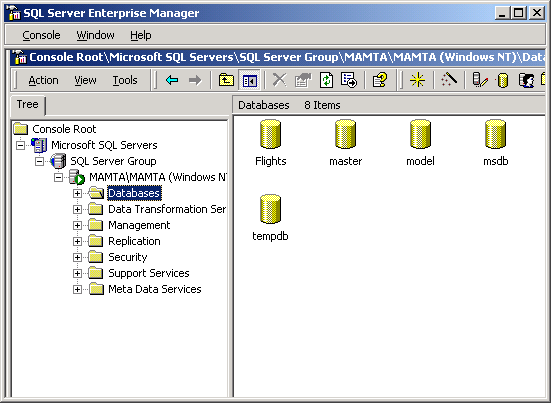
SQL Server Query

Analyzer

zer



Enterprise Manager window



What is a Query?



SQL Server



Query

Data in Tables

User

*Initiates*

*A request or a*

*question regarding*

*Answers the query – returns the result set*

*Processed by*

### What is .NET?

.NET is a catchall term that embraces Microsoft's core strategy, plans, and vision for the near future. At the heart of this strategy is the .NET Framework, which provides the core technology that underpins it all ASP.NET is just one of the several components that are present in the Framework.

.NET is designed to help solve several fundamental problems faced by programmers:

* Reduces the hard work involved in building large, reliable applications
* Allows programmers to unify two kinds of architectures – applications that run locally on a machine and applications that are accessed over the Web.
* Reduces overheads associated with programming frameworks – you don't need to write complex code with complicated languages to get an impressive performance out of .NET programs
* Allows programmers in different languages to work together on an application
* It has been designed with the view to accommodate various end-user tools, including desktops, PDAs, and cell phones

To sum it up, .NET provides an easier, and thus faster and cheaper way to get efficient programs into the hands of the users.

We can break down the entire .NET Framework into several core concepts:

MS Intermediate Language (MSIL): All the code written in .NET is compiled into a more abstract, trimmed-down form before it is executed. A programmer can use any .NET

language to write the code including Visual Basic (VB), C#, Jscript, and about 20 others. The result is then compiled to MSIL, the common language of .NET.

Common Language Runtime (CLR): This runtime is a complex system responsible for executing the MSIL code on the computer. It takes care of all the nitty-gritty tasks involved in talking to Windows and the Internet Information Services (IIS) server.

.NET Framework class libraries: These code libraries contain a mass of tremendously useful functionality, which we can very easily bolt onto our own applications to make complex tasks much more straightforward.

.NET languages: These are all the programming languages that conform to certain specific structural requirements as defined by the Common Language Specification (CLS) and can be compiled to MSIL. You can develop in any of the languages, such as C# or Visual Basic .NET, without any restrictions. This gives you the liberty to develop applications constructed from more than one of these languages.

ASP.NET: This module of code extends the IIS so that it can implement the .NET Framework for Web pages.

a special format, the MSIL. Some optimization can be done as part of this process, since the MSIL's structure doesn't have to be as easily human-readable as our original code. However, no machine-specific optimization is done. Thus, MSIL has the benefits of general optimization and of portability to any .NET server.

When we execute this page compiled to MSIL (when the user requests an ASP.NET page), we pass our code from MSIL to the CLR, another cornerstone of the .NET Framework. The CLR uses another compiler – the JIT (Just-In-Time) compiler – to compile to true machine code and make any last minute machine-specific optimizations to the program, so that it can run as quickly as possible on the local machine. Most importantly, MSIL is not machine-specific, so we can execute it on any machine that has the CLR installed. In essence, once we've written and compiled some .NET code, we can copy it to any machine with the CLR installed and executes it there. While the CLR is currently only compatible with Windows

(9x, NT, 2000, and XP versions), moves are already afoot to build versions for other operating systems.

MSIL and the CLR together give us the best of both words : the structural optimization of pre-compiled code along with the portability of interpreted code

objects. .NET is an object-oriented environment. An object is self-contained and offers a set of functions to the rest of your code. Writing code in objects and using pre-written objects has some important benefits:

* We avoid rewriting code for multiple uses.
* Objects allow us to buy functionality that may be beyond our ability or resources to develop.
* .NET objects are standardized, which means other programmers can easily discover and use an object's functionality.
* Objects can be written in any .NET-compliant language.

Thus, objects are a way to organize code so that programming is more efficient.

To be able to use the classes contained within a namespace, you need to import the namespace first. We can import these classes into our ASP.NET pages by simply adding a directive to the top of the file (before any HTML and before the language tag). For example, to make use of all the classes defined in the System.Drawing namespace, we just say:

<%@ Import Namespace= "System.Drawing" %>

Namespaces are used by .NET to group together classes in functionally similar

groups. Namespaces are not unique to .NET and are found in many non-.NET

languages as well.

#### Features of the Visual Studio .NET Environment

* Solution Explorer: Contains the root Solution and any sub-projects represented in a hierarchical (tree) arrangement. Because we only created a project, a root solution was created automatically. This is where you can double-click files to open them, right-click to rename them, or select different views on the same file.
* Toolbox: Contains all the controls you'll need to use to create Web pages – simply drag and drop, or double-click, to add each control.
* Server Explorer: Clicking the tab for the Server Explorer opens up another hierarchical tree representation of the server – you can use this to create connections to databases and view data held in these databases.
* File Selector Tabs: Contains a corresponding tab for each open file – simply click to make the selected file visible.
* Properties Pane: Contents change dynamically to reflect the properties available to the currently selected object. These properties change dynamically. The properties for the Web page itself are visible. Try clicking on items in the Server Explorer to view properties on each item in there. Later on, you'll see how to view properties on controls and edit them in this pane.
* HTML / Design View: Switch between the Design Surface or HTML view of a web page, similar to the behavior seen in Web Matrix. Code view can be displayed by right clicking on a page in server explorer and clicking View Code.
* Design Surface: In Design view, you can drag and drop controls from the toolbox onto the page. By default, elements are placed using absolute positioning (wherever you place them, they stay, and have x and y coordinates to place them on a page). Normally, we use Flow view (like in Web Matrix by default) to add controls to the top left of a page and work downwards, then use tables to arrange elements.

# ASP.NET Introduction

ASP.NET is a powerful and flexible technology for creating dynamic Web pages. It's a convergence of two major Microsoft technologies, Active Server Pages (ASP) and the .NET Framework. Active Server Pages, or ASP (or classic ASP as it's often referred to), is a relative old-timer on the Web computing circuit and has provided a sturdy, powerful, and effective way of building dynamic Web pages for seven years or so now. The .NET Framework, on the other hand, is a whole suite of technologies designed by Microsoft with the aim of revolutionizing the way in which all program development takes place and the way companies carry out business. ASP.NET is a way of creating dynamic Webpages while making use of the innovations present in the .NET Framework.

ASP.NET is the new offering for Web developers from the Microsoft .It is not simply the next-generation of ASP; in fact, it is a completely re-engineered and enhanced technology that offers much, much more than traditional ASP and can increase productivity significantly.

Active Server Pages (ASP)

Microsoft Active Server Pages (ASP) is a server-side scripting technology. ASP is a technology that Microsoft created to ease the development of interactive Web applications. With ASP you can use client-side scripts as well as server-side scripts. Maybe you want to validate user input or access a database. ASP provides solutions for transaction processing and managing session state. Asp is one of the most successful language used in web development.

Introducing ASP.NET

ASP.NET was developed in direct response to the problems that developers had with classic ASP. Since ASP is in such wide use, however, Microsoft ensured that ASP scripts execute without modification on a machine with the .NET Framework (the ASP engine, ASP.DLL, is not modified when installing the .NET Framework). Thus, IIS can house both ASP and ASP.NET scripts on the same machine

Advantages of ASP.NET

1. Separation of Code from HTML: To make a clean sweep, with ASP.NET you have the ability to completely separate layout and business logic. This makes it much easier for teams of programmers and designers to collaborate efficiently.

##### 2. Support for compiled languages: developer can use VB.NET and access features such as strong typing and object-oriented programming. Using compiled languages also means that ASP.NET pages do not suffer the performance penalties associated with interpreted code. ASP.NET pages are precompiled to byte-code and Just In Time (JIT) compiled when first requested. Subsequent requests are directed to the fully compiled code, which is cached until the source changes.

##### 3. Use services provided by the .NET Framework: The .NET Framework provides class libraries that can be used by your application. Some of the key classes help you with input/output, access to operating system services, data access, or even debugging. We will go into more detail on some of them in this module.

4. Graphical Development Environment: Visual Studio .NET provides a very rich development environment for Web developers. You can drag and drop controls and set properties the way you do in Visual Basic 6. And you have full IntelliSense support, not only for your code, but also for HTML and XML.

5. State management: To refer to the problems mentioned before, ASP.NET provides solutions for session and application state management. State information can, for example, be kept in memory or stored in a database. It can be shared across Web farms, and state information can be recovered, even if the server fails or the connection breaks down.

6. Update files while the server is running: Components of your application can be updated while the server is online and clients are connected. The Framework will use the new files as soon as they are copied to the application. Removed or old files that are still in use are kept in memory until the clients have finished.`Configuration settings in ASP.NET are stored in XML files that you can easily read and edit. You can also easily copy these to another server, along with the other files that comprise your application.

### ASP.NET Overview

* ASP.NET provides services to allow the creation, deployment, and execution of Web Applications and Web Services
* Like ASP, ASP.NET is a server-side technology
* Web Applications are built using Web Forms. ASP.NET comes with built-in Web Forms controls, which are responsible for generating the user interface. They mirror typical HTML widgets like text boxes or buttons. If these controls do not fit your needs, you are free to create your own user controls.
* Web Forms are designed to make building web-based applications as easy as building Visual Basic applications

###### ASP.NET Architecture

ASP.NET is based on the fundamental architecture of .NET Framework. Visual studio provides a uniform way to combine the various features of this Architecture.



Architecture is explained form bottom to top in the following discussion.

1. At the bottom of the Architecture is Common Language Runtime. NET Framework common language runtime resides on top of the operating system services. The common language runtime loads and executes code that targets the runtime. This code is therefore called managed code. The runtime gives you, for example, the ability for cross-language integration.

2. .NET Framework provides a rich set of class libraries. These include base classes, like networking and input/output classes, a data class library for data access, and classes for use by programming tools, such as debugging services. All of them are brought together by the Services Framework, which sits on top of the common language runtime.

3.ADO.NET is Microsoft’s ActiveX Data Object (ADO) model for the .NET Framework. ADO.NET is not simply the migration of the popular ADO model to the managed environment but a completely new paradigm for data access and manipulation.ADO.NET is intended specifically for developing web applications.

ASP.NET Applications and Configuration

Overview

Like ASP, ASP.NET encapsulates its entities within a web application. A web application is an abstract term for all the resources available within the confines of an IIS virtual directory. For example, a web application may consist of one or more ASP.NET pages, assemblies, web services configuration files, graphics, and more. In this section we explore two fundamental components of a web application, namely global application files (Global.asax) and configuration files (Web.config).

Global.asax

Global.asax is a file used to declare application-level events and objects. Global.asax is the ASP.NET extension of the ASP Global.asa file. Code to handle application events (such as the start and end of an application) reside in Global.asax. Such event code cannot reside in the ASP.NET page or web service code itself, since during the start or end of the application, its code has not yet been loaded (or unloaded). Global.asax is also used to declare data that is available across different application requests or across different browser sessions. This process is known as application and session state management.

The Global.asax file must reside in the IIS virtual root. Remember that a virtual root can be thought of as the container of a web application. Events and state specified in the global file are then applied to all resources housed within the web application. If, for example, Global.asax defines a state application variable, all .aspx files within the virtual root will be able to access the variable.

Like an ASP.NET page, the Global.asax file is compiled upon the arrival of the first request for any resource in the application. The similarity continues when changes are made to the Global.asax file; ASP.NET automatically notices the changes, recompiles the file, and directs all new requests to the newest compilation. A Global.asax file is automatically created when you create a new web application project in the VS.NET IDE.

Application Directives

Application directives are placed at the top of the Global.asax file and provide information used to compile the global file. Three application directives are defined, namely Application, Assembly, and Import. Each directive is applied with the following syntax:

<%@ appDirective appAttribute=Value ...%>

Web.config

In ASP, configuration settings for an application (such as session state) are stored in the IIS metabase. There are two major disadvantages with this scheme. First, settings are not stored in a human-readable manner but in a proprietary, binary format. Second, the settings are not easily ported from one host machine to another.(It is difficult to transfer information from an IIS’s metabase or Windows Registry to another machine, even if it has the same version of Windows.)

Web.config solves both of the aforementioned issues by storing configuration information as XML. Unlike Registry or metabase entries, XML documents are human-readable and can be modified with any text editor. Second, XML files are far more portable, involving a simple file transfer to switch machines.

Unlike Global.asax, Web.config can reside in any directory, which may or may not be a virtual root. The Web.config settings are then applied to all resources accessed within that directory, as well as its subdirectories. One consequence is that an IIS instance may have many web.config files. Attributes are applied in a hierarchical fashion. In other words, the web.config file at the lowest level directory is used.

Since Web.config is based on XML, it is extensible and flexible for a wide variety of applications. It is important, however, to note that the Web.config file is optional. A default Web.config file, used by all ASP.NET application resources, can be found on the local machine at:

%winroot%Microsoft.NetFrameworkversionCONFIGmachine.config

INTERNET INFORMATION SERVER

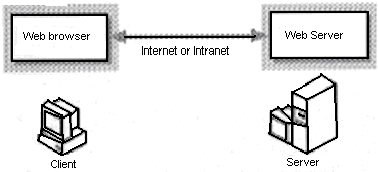
Microsoft Internet information server is a web server, which enables us to publish information on a corporate intranet or on the Internet. Internet information server transmits information by using the hypertext transfer protocol (HTTP). Internet information server can also be configured to provide the file transfer protocol (FTP) and gopher services. The FTP service enables users to transfer files to and from your website. The gopher services uses a menu driven protocols for locating documents. The gopher protocols has largely superseded by the HTTP protocol.

The creative possibilities of what us can offer on an Internet information server Web site are endless. Some familiar are to:

* Publish a home page on an Internet for your business featuring a newsletter, sales information, or employment opportunities.
* Publish a catalog and take orders from the customers.
* Publish interactive programs
* Provide your remote sales force easy access to your sales database.
* Use an order-tracking database.

### HOW DOES INTERNET INFORMATION SERVER WORKS

The Web is fundamentally a system of requests and responses. Web browser’s requests information by sending a URL to a Web server. Web server responds by returning a hypertext markup language (HTML) page.



### Figure 1

The HTML page can be a static page that has already been formatted and stored in the Web site, a page that the server dynamically creates in response to information provide by the user, or a page that lists the available files and the folders on the Web site.

## DIRECTORY LISTING:-

If users might send queries without specifying a particular file, user can either create a default document for a web site or for a particular directory, or user can configure your server for directory listing.

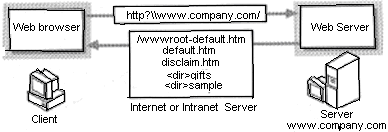


Figure 5

Rather then using directory listing, us can provide a default document.

USE OF THE INTERNET INFORMATION SERVER

Internet information server is flexible enough to perform many important functions for your organization. It is scalable from supporting a single-server site to supporting large multi-server installations. For example, www.microsoft.com and www.msn.com are among the busiest web sites on the Internet today, and both use multiple servers running Microsoft Internet Information Server.

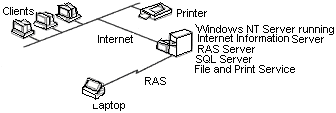
One of the primary factors that determines the configuration and use of Internet Information Server is whether employees on your intranet will use it internally or if it will be connected to the Internet.

The following scenarios are intended to help us understand the range of possibilities for using Internet Information Server.

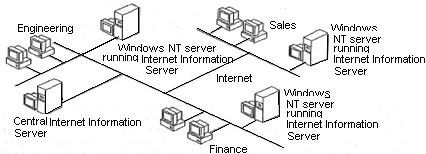
INTRANET SCENARIOS:-

Internet Information Server integrates well into almost any existing environment. Because Internet Information Server integrates Windows NT security and networking, us can often add the software to an existing computer and use existing user accounts. It is not necessary to use a dedicated computer to run Internet Information Server.

For example in a small workgroup us can add Internet Information Server to an existing file and print server. The workgroup web server can host personal web style pages customized workgroup applications, serve as an interface to the workgroups structured query language (SQL) data bases, or use remote access service (RAS) to provide dial up access to the work groups resources and remote sites.

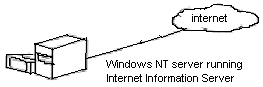


In a larger business with multiple departments or workgroups, each department might run Internet Information Server on an existing file server for workgroup specific information. A central information server might be used for the company wide information, such as an employ manual or company directory.



INTERNET SCENARIOS

Internet Information Server can function as a simple dedicated web server on the

Internet,as shown in the figure.  
  Figure 8

In larger sites us can provides access from your internal network to the Internet information server, allowing the employees to brows the server or to use the authoring tools, such as Microsoft front page, to create content for your server.

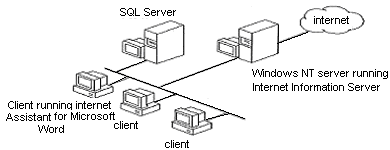


Figure 9

Internet Information Servers integration with all the Windows NT services can also create servers with multiple functions for examples; a company with sites in different parts of the world can use Internet Information Server to provide communication between sites, with the added flexibility of Internet access. You can even add RAS to an Internet Information Server to provide the dialup access to your intranet or the Internet.

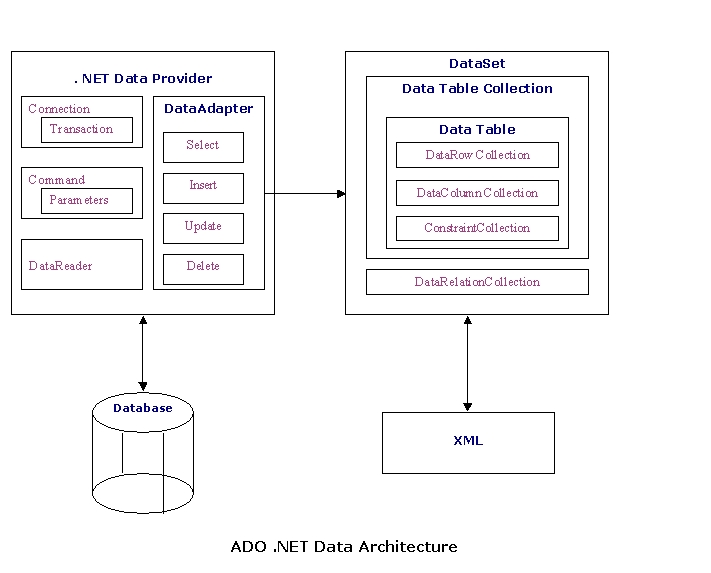
 Figure 10

ADO .NET:-

Most applications need data access at one point of time making it a crucial component when working with applications. Data access is making the application interact with a database, where all the data is stored. Different applications have different requirements for database access. ASP.NET uses ADO .NET (Active X Data Object) as it's data access and manipulation protocol which also enables us to work with data on the Internet.

Data access with ADO.NET can be summarized as follows:

A connection object establishes the connection for the application with the database. The command object provides direct execution of the command to the database. If the command returns more than a single value, the command object returns a DataReader to provide the data. Alternatively, the DataAdapter can be used to fill the Dataset object. The database can be updated using the command object or the DataAdapter.



Component classes that make up the Data Providers

The Connection Object

The DataReader Object

The DataReader object provides a forward-only, read-only, connected stream recordset from a database. Unlike other components of the Data Provider, DataReader objects cannot be directly instantiated. Rather, the DataReader is returned as the result of the Command object's ExecuteReader method. The SqlCommand.ExecuteReader method returns a SqlDataReader object, and the OleDbCommand.ExecuteReader method returns an OleDbDataReader object. The DataReader can provide rows of data directly to application logic when you do not need to keep the data cached in memory. Because only one row is in memory at a time, the DataReader provides the lowest overhead in terms of system performance but requires the exclusive use of an open Connection object for the lifetime of the DataReader.

The DataAdapter Object

The DataAdapter is the class at the core of ADO .NET's disconnected data access. It is essentially the middleman facilitating all communication between the database and a DataSet. The DataAdapter is used either to fill a DataTable or DataSet with data from the database with it's Fill method. After the memory-resident data has been manipulated, the DataAdapter can commit the changes to the database by calling the Update method. The DataAdapter provides four properties that represent database commands:

SelectCommand

InsertCommand

DeleteCommand

UpdateCommand

When the Update method is called, changes in the DataSet are copied back to the database and the appropriate InsertCommand, DeleteCommand, or UpdateCommand is executed.

### 

### Microsoft SQL SERVER 2008

Microsoft® SQL Server™ 2008 is designed to work effectively as:

* A central database on a server shared by many users who connect to it over a network. The number of users can range from a handful in one workgroup, to thousands of employees in a large enterprise, to hundreds of thousands of Web users.
* A desktop database that services only applications running on the same desktop.
* Server Database Systems

Server-based systems are constructed so that a database on a central computer, known as a server, is shared among multiple users. Users access the server through an application:

* In a multitier system, such as Windows® DNA, the client application logic is run in two or more locations.
* A thin client is run on the user's local computer and is focused on displaying results to the user.
* The business logic is located in server applications running on a server. Thin clients request functions from the server application, which is itself a multithreaded application capable of working with many concurrent users. The server application is the one that opens connections to the database server.
* This is a typical scenario for an Internet application. For example, a multithreaded server application can run on a Microsoft® Internet Information Services (IIS) server and service thousands of thin clients running on the Internet or an intranet. The server application uses a pool of connections to communicate with one or more instances of SQL Server 2008. The instances of SQL Server 2008 can be on the same computer as IIS, or they can be on separate servers in the network.
* In a two-tier client/server system, users run an application on their local computer, known as a client application, that connects over a network to an instance of SQL Server 2008 running on a server computer. The client application runs both business logic and the code to display output to the user, so this is sometimes referred to as a thick client.
* Advantages of Server Database System

Having data stored and managed in a central location offers several advantages:

* Each data item is stored in a central location where all users can work with it.
* Business and security rules can be defined one time on the server and enforced equally among all users.
* A relational database server optimizes network traffic by returning only the data an application needs.
* Hardware costs can be minimized.
* Maintenance tasks such as backing up and restoring data are simplified because they can focus on the central server.
* Advantages of SQL Server 2008 as a Database Server

Microsoft SQL Server 2008 is capable of supplying the database services needed by extremely large systems. Large servers may have thousands of users connected to an instance of SQL Server 2008 at the same time. SQL Server 2008 has full protection for these environments, with safeguards that prevent problems, such as having multiple users trying to update the same piece of data at the same time. SQL Server 2008 also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users. Extremely large Internet sites can partition their data across multiple servers, spreading the processing load across many computers, and allowing the site to serve thousands of concurrent users.

Multiple instances of SQL Server 2008 can be run on a single computer. For example, an organization that provides database services to many other organizations can run a separate instance

of SQL Server 2008 for each customer organization, all on one computer. This isolates the data for each customer organization, while allowing the service organization to reduce costs by only having to administer one server computer.

SQL Server 2008 applications can run on the same computer as SQL Server 2008. The application connects to SQL Server 2008 using Windows Interprocess Communications (IPC) components, such as shared memory, instead of a network. This allows SQL Server 2008 to be used on small systems where an application must store its data locally.

The illustration above shows an instance of SQL Server 2008 operating as the database server for both a large Web site and a legacy client/server system.

The largest Web sites and enterprise-level data processing systems often generate more databases processing than can be supported on a single computer. In these large systems, the database services are supplied by a group of database servers that form a database services tier. SQL Server 2008 supports a mechanism that can be used to partition data across a group of autonomous servers. Although each server is administered individually, the servers cooperate to spread the database-processing load across the group.

* What's New in Microsoft SQL Server 2008

Microsoft® SQL Server™ 2000 extends the performance, reliability, quality, and ease-of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2008 includes several new features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce applications.

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2008 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component.

The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2008 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services.

* Relational Database Enhancements

Microsoft® SQL Server™ 2000 introduces several server improvements and new features:

* 1. XML Support
  2. Federated Database Servers
  3. User-Defined Functions
  4. Indexed Views
  5. New Data Types
  6. INSTEAD OF and AFTER Triggers
  7. Cascading Referential Integrity Constraints
  8. Full-Text Search Enhancements
  9. Multiple Instances of SQL Server
  10. Index Enhancements
  11. Failover Clustering Enhancements
  12. Net-Library Enhancements
  13. 64-GB Memory Support
  14. Distributed Query Enhancements
  15. Updatable Distributed Partitioned Views
  16. Kerberos and Security Delegation
  17. Backup and Restore Enhancements
  18. Scalability Enhancements for Utility Operations
* Client Components

Clients do not access Microsoft® SQL Server™ 2000 directly; instead, clients use applications written to access the data in SQL Server. SQL Server 2008 supports two main classes of applications:

* Relational database applications that send Transact-SQL statements to the database engine; results are returned as relational result sets.
* Internet applications that send either Transact-SQL statements or XPath queries to the database engine; results are returned as XML documents.
* Relational Database APIs

SQL Server 2008 provides native support for two main classes of database APIs:

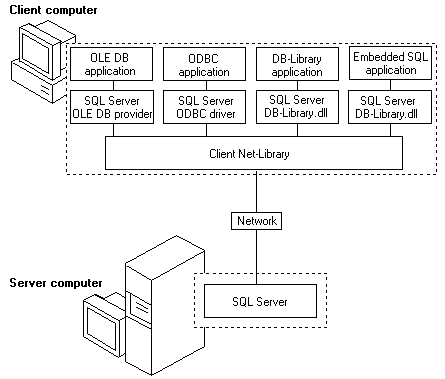
* OLE DB SQL Server 2008 includes a native OLE DB provider. The provider supports applications written using OLE DB, or other APIs that use OLE DB, such as ActiveX Data Objects (ADO). Through the native provider, SQL Server 2008 also supports objects or components using OLE DB, such as ActiveX, ADO, or Windows DNA applications.
* ODBC SQL Server 2008 includes a native ODBC driver. The driver supports applications or components written using ODBC, or other APIs using ODBC, such as DAO, RDO, and the Microsoft Foundation Classes (MFC) database classes.
* Additional SQL Server API Support

SQL Server 2008 also supports:

* DB-Library
* Embedded SQL

Client Communications

The Microsoft OLE DB Provider for SQL Server 2008, the SQL Server 2008 ODBC driver, and DB-Library are each implemented as a DLL that communicates to SQL Server 2008 through a component called a client Net-Library.

.

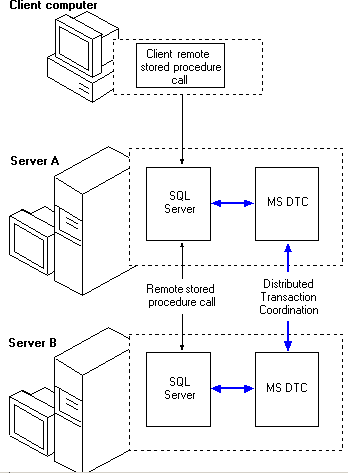
* MS DTC Service

The Microsoft Distributed Transaction Coordinator (MS DTC) is a transaction manager that allows client applications to include several different sources of data in one transaction. MS DTC coordinates committing the distributed transaction across all the servers enlisted in the transaction.

An installation of Microsoft® SQL Server™ can participate in a distributed transaction by:

* Calling stored procedures on remote servers running SQL Server.

Automatically or explicitly promoting the local transaction to a distributed transaction and enlist remote servers in the transaction.



* Using Data Types

Objects that contain data have an associated data type that defines the kind of data (character, integer, binary, and so on) the object can contain. The following objects have data types:

* Columns in tables and views.
* Parameters in stored procedures.
* Variables.
* Transact-SQL functions that return one or more data values of a specific data type.
* Stored procedures that have a return code, which always has an integer data type.

Assigning a data type to an object defines four attributes of the

Constraints allow you to define the way Microsoft® SQL Server™ 2000 automatically enforces the integrity of a database. Constraints define rules regarding the values allowed in columns and are the standard mechanism for enforcing integrity.

Classes of Constraints

SQL Server 2008 supports five classes of constraints.

* NOT NULL specifies that the column does not accept NULL values.
* CHECK constraints enforce domain integrity by limiting the values that can be placed in a column.
* UNIQUE constraints enforce the uniqueness of the values in a set of columns.
* PRIMARY KEY constraints identify the column or set of columns whose values uniquely identify a row in a table.
* FOREIGN KEY constraints identify the relationships between tables.
  + NO ACTION specifies that the deletion fails with an error.
  + CASCADE specifies that all the rows with foreign keys pointing to the deleted row are also deleted.
* SQL Server Enterprise Manager

SQL Server Enterprise Manager is the primary administrative tool for Microsoft® SQL Server™ 2000 and provides a Microsoft Management Console (MMC)–compliant user interface that allows users to:

* Define groups of servers running SQL Server.
* Register individual servers in a group.
* Configure all SQL Server options for each registered server.
* Create and administer all SQL Server databases, objects, logins, users, and permissions in each registered server.
* Define and execute all SQL Server administrative tasks on each registered server.
* Design and test SQL statements, batches, and scripts interactively by invoking SQL Query Analyzer.
* Invoke the various wizards defined for SQL Server.
* SQL Server Query Analyzer

SQL Server Query Analyzer is a graphical User Interface for designing and testing Transact-SQL statements, batches, and scripts interactively. SQL Server Query Analyzer offers the following features:

* Free-form text editor for keying in Transact-SQL statements.
* Color coding of Transact-SQL syntax to improve the readability of complex statements
* Results presented in either a grid or a free form text window.
* Graphical diagram of the show plan information showing the logical steps built into the execution plan of a Transact-SQL statement.
* Index tuning wizard to analyze a Transact-SQL statement and the tables it references to see if adding additional indexes will improve the performance of the query.
* Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL. Standards for SQL have been defined by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO).Most modern DBMS products support the Entry Level of SQL-92, the latest SQL standard (published in 1992).

### A brief introduction to the SQL language is presented here.

SQL is the universal language used for data base management. SQL commands are specialized set of programming commands that enable

* Retrieval of data from one or more databases.
* Manipulate data in one or more tables by inserting, deleting, or updating records.
* Obtain summary information about the data in the tables, such as maximum, minimum, average values etc.

SQL is a declarative language i.e. we just need to specify what we want and not how to do it. SQL statements can be basically divided into three types of statements.

* DDL (data definition language) statements are used to define the database.
* DML (data manipulation language) statements are used to access the database.
* DCL (data control language) statements are used to protect the database from the unauthorized users, concurrent or multi-user transactions, power failures etc.

Typically SQL statements consist of three sections

* Parameter declarations are optional parameter passed by the program code to the sql statement.
* Manipulative statements tell the database engine the process it will perform.
* Optional declarations define and filter conditions, groupings or sort that must be applied to the data being processed.

Thus the general syntax of a standard SQL statement will be

[Parameters] manipulated statements [Options]

There are four basic commands, which are used in almost all queries to a database using SQL.

SELECT Command

The SELECT statement is used to select data from a table. The tabular result is stored in a result table (called the result set). It can perform both of the relational algebra operations selection and projection.

##### A Typical SELECT Statement

SELECT [predicate] field list

FORM Table Expression

[WHERE RowCondition]

[GROUP BY GroupByCriteria]

[HAVING GroupCriteria]

[ORDER BY OrderByCriteria]

INSERT Command

The INSERT INTO statement is used to insert new rows into a table.

A Typical INSERT Statement

INSERT INTO target [(FieldName,….)]

VALUES (Value1,….)

DELETE Command

The DELETE Command statement is used to delete rows in a table.

A Typical DELETE Statement

DELETE FROM table\_name

WHERE column\_name = some\_value

UPDATE Command

###### The UPDATE statement is used to modify the data in a table

A Typical UPDATE Statement

UPDATE table\_name

SET column\_name = new\_value

WHERE column\_name = some\_value

* Importing & Exporting Data

Importing data is the process of retrieving data from sources external to Microsoft SQL Server, for example, an ASCII text file, and inserting the data into SQL Server tables. Exporting data is the process of extracting data from SQL Server into some user-specified format, for example, copying the contents of a SQL Server table to a Microsoft Access database.

SQL Server provides tools for importing and exporting data to and from a variety of data sources including text files, ODBC data sources (such as Oracle databases), OLE DB data sources (such as other servers running SQL Server), ASCII text files, and Excel spreadsheets.

* Backing Up and Restoring Databases

The back-up and restore component of Microsoft SQL Server provides an important safeguard for protecting critical data stored in SQL Server databases. Backing up and restoring a database allows for the complete restoration of data over a wide range of potential system problems:

* Media failure

If one or more of the disk drives holding a database fail, you are faced with a complete loss of data unless you can restore an earlier copy of the data.

* User errors

If a user or application either unintentionally or maliciously makes a large number of invalid modifications to data , the best way to deal with the problem may be to restore the data to a point in time before the modifications were made.

* permanent loss of a server
* If a server is disabled permanently, or a site is lost to a natural disaster, you may need to activate a warm standby server or restore a copy of a database to another server.

1. Backing up a Database

Backing up a database makes a copy of a database, which can be used to restore the database if it is lost. Backing up database copies everything in the database, including any needed portions of transaction log.

The transaction log is a serial record of all the modifications that have occurred in a database, and which transaction performed each modification. The transaction log is used during recovery operations to roll forward completed transactions, and roll back (undo) uncompleted transactions. Backing up a transaction log was last backed up.

A backup operates like a fuzzy snapshot taken of a database or transaction log:

* A database backup records the complete state of the data in the database at the time the backup operation completes.
* A transaction log backup records the state of the transaction log at the time the backup operation starts.

1. Restoring a Database

Restoring a database backup returns the database to the same state it was in when the backup was created. Any incomplete transactions in the database back up, (transactions that were not complete when the backup operation completed originally), are rolled back to ensure the database remains consistent.

* Permissions Validation

After a user has been authenticated and allowed to log into Microsoft SQL Server using their login, a user account is required in each database the user needs to access. Requiring a user account in each database prevents users from being able to connect to SQL Server and access all the databases on a server.

For example, if a server contains personnel database and recruiting database users who should be able to access the recruiting database but not the personnel database would have a user account created only in the recruiting database.

* Setting up Security Accounts

Each user needs to gain access to Microsoft SQL Server through a login account that establishes the ability to connect (authentication). This login then has to be mapped to a SQL Server user account used to control activities performed in the database (permission validation). Therefore, a single login is mapped to one user account created in each database the login has to access. If no user account exits in a database, the user can’t access the database even though the user may be able to connect to SQL Server.

The login is created in Microsoft Windows NT ® rather than in SQL Server. This login (A Windows NT user or group account) is then granted permission to connect to SQL.

The login is created within SQL Server.

* Application Security and Application Roles

The Security system in Microsoft SQL Server is implemented at the lowest level, the database itself. This is the best, most robust method for controlling user activities regardless of the application used to communicate with SQL Server. However, you may have situations when security controls must be customized to accommodate the special requirements of an individual application, essentially when dealing with complex databases and databases with large tables.

* Encryption

Encryption is a method for keeping sensitive information confidential by changing data into an unreadable form. Encryption ensures that data remains secure by keeping the information hidden from everyone, even if the encrypted data is viewed directly. Decryption is the process of changing encrypted data back into its original form so authorized users can view it.

Microsoft® SQL Server™ encrypts or can encrypt:

* Login and application role passwords stored in SQL Server.
* Any data sent between the client and the server as network packets.
* Stored procedure definitions.
* User-defined function definitions.
* View definitions.
* Trigger definitions.
* Default definitions.
* Rule definitions.
* SQL Server File Permissions

Microsoft SQL Server must create and access files to store databases, database backups, error logs, and so on. The SQL Server process must run in the context of a security account that has the necessary permissions to create and access these files, whether these files exist on the local computer or a network drive on a remote computer. The security account SQL Server uses depends on the method used to start SQL Server.

DEVELOPMENT TOOL

Visual Studio .NET :- 

Microsoft Visual C# is just one of the languages that uses Visual Studio .NET as its development environment. Other programming languages supplied by Microsoft that use Visual Studio .NET include Visual Basic .NET and Visual C++ .NET. In addition, companies other than Microsoft are supplying compilers for Visual Studio .NET, which will enable you to develop solutions that include Eiffel, COBOL, and other languages. All programming using these languages can take advantage of the same set of tools and features offered by Visual Studio .NET, including all of the designers and tool windows that are part of the integrated development environment (IDE) as well as the integrated help system.

|  |
| --- |
| [http://upload.wikimedia.org/wikipedia/en/thumb/6/60/MSVisual_Studio.png/180px-MSVisual_Studio.png](http://en.wikipedia.org/wiki/File:MSVisual_Studio.png) |
|  |

The Start Page

The Visual Studio .NET Start Page, shown in Figure, provides a home base for obtaining information and services that extend beyond your machine. Many of the tabs available to you on the Start Page require an Internet connection. These tabs provide late-breaking information about Visual Studio .NET, provide links to new downloads, and enable you to host Web services with just a couple of mouse clicks.

Specifically, the Visual Studio .NET Start Page contains the following tabs:

Get Started: The Get Started tab is displayed initially and includes a list of solutions that you’ve worked with recently. When you first launch Visual C# .NET, the list of solutions will be empty. Later the list will contain links to your most recent projects. This tab also includes a link for reporting Visual Studio .NET bugs. If you have a connection to the Internet, Visual Studio .NET will check for service packs and add a link to the update page when a service pack is available.

What’s New:  What’s New tab is a Web page that has the latest updates and information about Visual Studio .NET. This tab includes additional up-to-date content that’s downloaded from the Microsoft Web site if you have an Internet connection.

Online Community: The Online Community tab has a selection of newsgroups and Web sites for Visual Studio .NET developers.

Headlines: The Headlines tab contains information from the Microsoft Developer Network (MSDN) Web site, including the latest features, technical information, news, and links to relevant knowledge base articles. To use this tab, you must have a connection to the Internet.

Search Online: The Search Online tab provides access to online searching of the MSDN database. To use this tab, you must have a connection to the Internet.

Downloads: The Downloads tab provides links to download and code samples that are related to developing with Visual C# .NET and Visual Studio .NET. To use this tab, you must have a connection to the Internet.

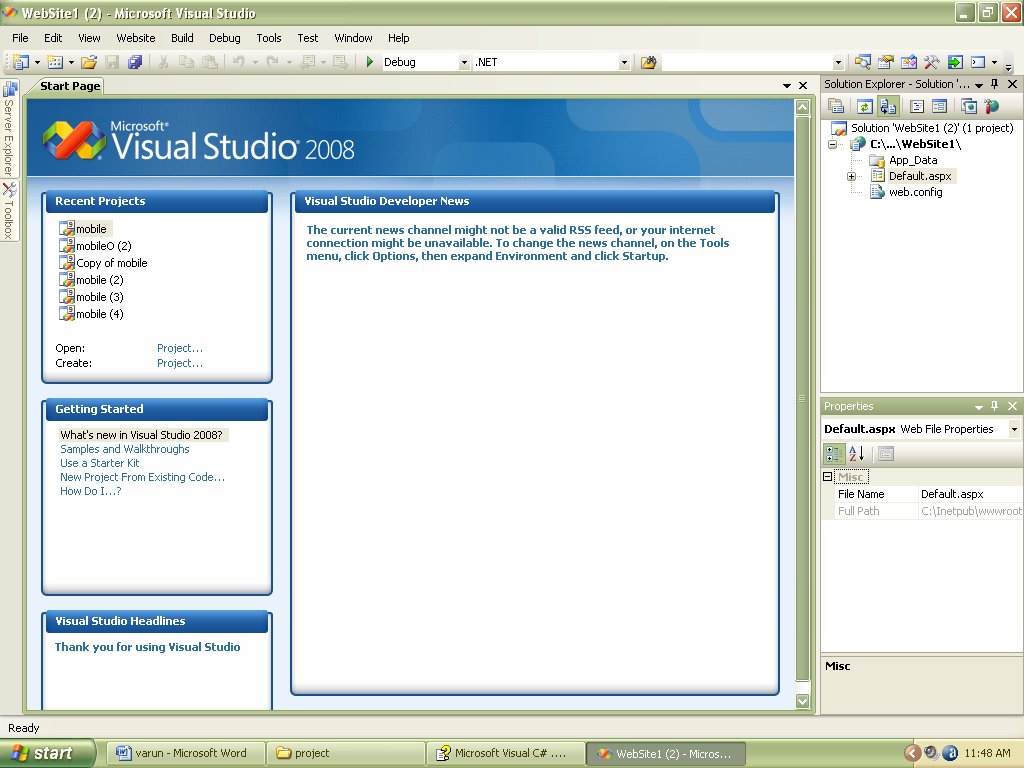
XML Web Services: The XML Web Services tab allows you to search for XML Web services you want to use in your current project. To use this tab, you must have a connection to the Internet.

Web Hosting:  The Web Hosting tab provides information about Web hosting options that are available to you as a .NET developer using Visual C# .NET. Visual Studio .NET includes one-click hosting, a simplified way to host your applications. To use this tab, you must have a connection to the Internet.

My Profile: The My Profile tab is used to configure the Visual Studio .NET user interface according to predefined profile elements.

Visual Studio .NET Windows

Visual Studio .NET has a large number of windows, toolbars, and Toolbox windows that you’ll use when developing your Visual C# applications. The environment is completely customizable, and the location and appearance of most windows can be easily modified to suit your needs. The default Visual Studio .NET layout is shown in Figure



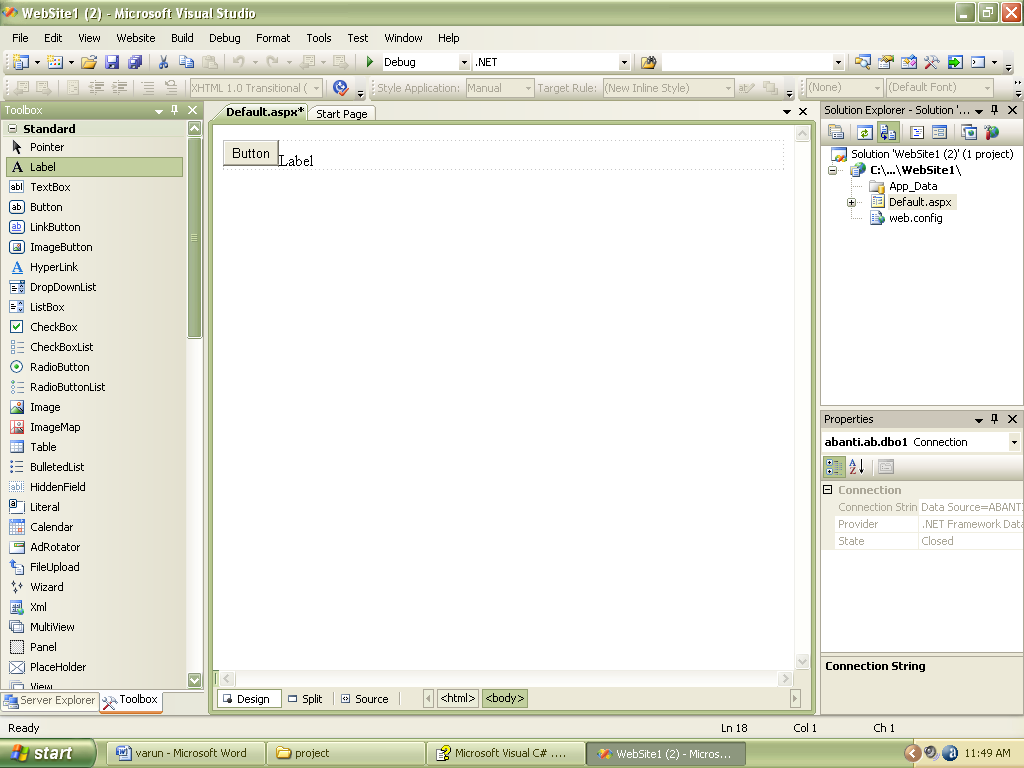
* Solution Explorer

Solution Explorer displays a tree of the current Visual Studio .NET solution. Using Solution Explorer, you can browse through all the projects that make up the current solution, as well as the files that belong to each project.

Double-clicking a project file will open the file for editing. Opening a file will change the menu and toolbar items that are available. For example, if you open an XML file, a top-level menu item for XML operations will be added to the main menu. Right-clicking on any element in Solution Explorer will display a shortcut menu with actions that you can perform on that element. For example, the shortcut menu for the solution icon allows you to perform tasks such as adding a new project to the solution, whereas the shortcut menu for a project enables you to add new items to the project and perform other project-related activities.

* Properties

The Properties window is used to declaratively set properties for different elements of your solution. The contents of this window vary depending on the type of item you’re currently working with. If you click an icon in Solution Explorer, properties for the selected item will be displayed. If you’re working with user interface controls or forms, many of the control and form properties can be set through this window. Likewise, if you’re working with an HTML or XML document, object model properties for these documents can be set in this window.

* Server Explorer

Server Explorer provides access to system services that are available on your machine as well as on other machines on your network. Normally, Server Explorer is tucked away under the edge of the Visual Studio .NET window, with just a small icon visible. If you position the mouse pointer over the Server Explorer icon, the window expands to display a list of servers available on your network. To lock the Server Explorer window in place, click the pin icon in the upper-right; this will prevent the window from auto-hiding until the pin icon is clicked again.

Server Explorer provides easy access to event logs, databases, performance counters, and other system services. Server Explorer is more than just a console for viewing information; you can drag objects from Server Explorer into your project, and Visual C# .NET will automatically generate code to make use of the new objects in your project. For example, to add a database connection to a Windows application, you can simply drag a database icon from Server Explorer to an open form in the Visual Studio .NET Windows Forms Designer.

* Toolbox

The Toolbox window contains items that can be easily added to your projects. Each tool category is located on a separate tab in the Toolbox window; clicking a tab displays the items in that category. For example, in Figure, the Web Forms tab is open, showing a number of controls and elements that are useful when you’re creating a Web form.

The Web Forms tab of the Toolbox window, which provides easy access to commonly used items.

Visual Studio .NET will populate the Toolbox window with tool categories that are relevant to your current project. For example, a Web application will have Toolbox items that provide data access and controls for Web Forms. You can control the behavior of the Toolbox window through a shortcut menu that appears when you right-click in the Toolbox window.

* File Selector Tabs

Contains a corresponding tab for each open file – simply click to make the selected file visible.

* b

Switch between the Design Surface or HTML view of a web page, similar to the behavior seen in Web Matrix. Code view can be displayed by right clicking on a page in server explorer and clicking View Code.

* Design Surface

In Design view, you can drag and drop controls from the toolbox onto the page. By default, elements are placed using absolute positioning (wherever you place them, they stay, and have x and y coordinates to place them on a page). Normally, we use Flow view (like in Web Matrix by default) to add controls to the top left of a page and work downwards, then use tables to arrange elements.

Risk Analysis:-

Uncertainty, which is constantly present in our daily lives, frequently impacts our decisions and actions.  When we talk about risk, we normally mean the chance that some undesirable impact will occur.  Hence, we normally seek to avoid or minimize risk.  If there is a chance of rain, and we don't want to get wet, we may choose to stay indoors -- avoiding that risk -- or we may take an umbrella to minimize the impact of rain upon us. Uncertainty can impact our decisions and actions in desirable as well as undesirable ways.  In risk analysis we usually focus on what can go wrong -- the outcomes that represent loss or damage -- although an effective analysis will also help us understand what can go right as well.

A risk assessment involves evaluating existing physical and environmental security and controls, and assessing their adequacy relative to the potential threats of the organization. A business impact analysis involves identifying the critical business functions within the organization and determining the impact of not performing the business function beyond the maximum acceptable outage. Types of criteria that can be used to evaluate the impact include: customer service, internal operations, legal/statutory and financial.

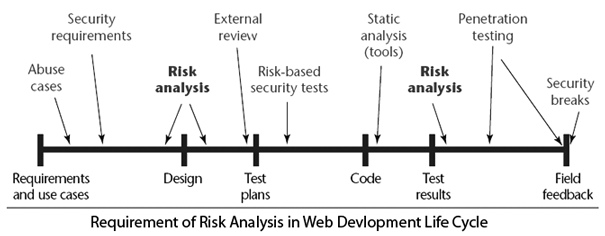
Our business processes depend heavily on technology and automated systems, and their disruption for even a few days could cause severe financial loss and threaten survival. The continued operations of this organization depend on management’s awareness of potential disasters, their ability to develop a plan to minimize disruptions of mission critical functions, and the capability to recover operations expediently and successfully. The risk analysis process provides the foundation for the entire recovery planning effort.

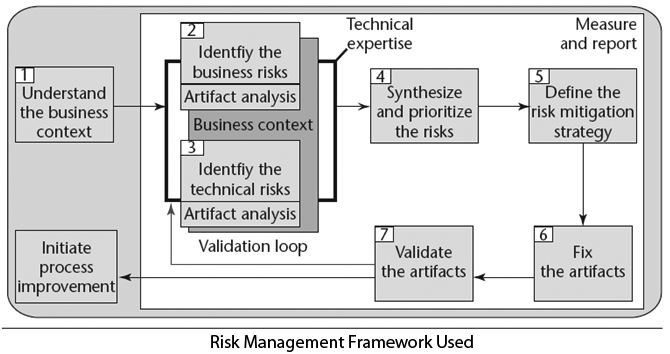
A primary objective of business recovery planning is to protect the organization in the event that all or part of its operations and/or computer services is rendered unusable. Each functional area of the organization should be analyzed to determine the potential risk and impact related to various disaster threats.

Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences are difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization. Regardless of the type of threat, the goals of business recovery planning are to ensure the safety of customers, employees and other personnel during and following a disaster.

Uncertainty can arise in several ways:

* If the quantity we'd like to know is a competing firm's planned product price, uncertainty arises from our lack of knowledge:  The price may be well known to that firm's employees, but it's unknown to us.
* If the quantity is market demand for products like ours, uncertainty arises from the complexity of the process:  Demand depends on economic factors, fashions and preferences, and our and other firms' actions -- and even if we knew all of these, we couldn't fully calculate their net impact on final demand.
* If the quantity is a material thickness in nanometers, uncertainty may arise from limits on our ability to measure this physical quantity.  We may also have limits on our ability to control fabrication of the material.
* Many processes that we want to model -- from the failure rate of an electronic component to the behavior of a macromolecule -- have inherent randomness for all intents and purposes.





Cost estimation:-

It is the responsibility of the project manager to make accurate estimations of effort and cost. This is particularly true for projects subject to competitive bidding where a bid too high compared with competitors would result in loosing the contract or a bid too low could result in a loss to the organization. This does not mean that internal projects are unimportant. From a project leaders estimate the management often decide whether to proceed with the project. Industry has a need for accurate estimates of effort and size at a very early stage in a project. However, when software cost estimates are done early in the software development process the estimate can be based on wrong or incomplete requirements. A software cost estimate process is the set of techniques and procedures that an organization uses to arrive at an estimate. An important aspect of software projects is to know the cost; the major contributing factor is effort.

**Why SCE is difficult and error prone ?**

Software cost estimation requires a significant amount of effort to perform it correctly.

SCE is often done hurriedly, without an appreciation for the effort required.

You need experience at developing estimates, especially for large projects.

Human bias i.e An Estimator is likely to consider how long a certain portion of the system would take, and then to merely extrapolate this estimate to the rest of the system, ignoring the non-linear aspects of software development.

**The causes of poor and inaccurate estimation**

imprecise and drifting requirements

new software projects are nearly always different form the last.

software practitioners don't collect enough information about past projects.

estimates are forced to match the resources available.

**ALGORITHMIC COST MODELS:-**

To date most work carried out in the software cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulae linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by calibrating the model to your specific development environment, which basically involves adjusting the weightings of the metrics. There are a variety of different models available, the best known are Boehm's COCOMO [BOEHM-81], Putman's SLIM, and Albrecht's' function points [ALBR-83]. On an initial instinct you might expect formal models to be advantageous for their 'off-the-shelf' qualities, but after close observation this is regarded as a disadvantage by cost estimators due to the additional overhead of calibrating the system to the local circumstances. However, the more time spent calibrating a formal model the more accurate the cost estimate should be. A distinct disadvantage of formal models is the inconsistency of estimates, [KEMERER] conducted a study indicating that estimates varied from as much as 85 - 610 % between predicated and actual values. Calibration of the model can improve these figures, However, formal models still produce errors of 50-100%.

In terms of the estimation process , nearly all algorithmic models deviate from the classical view of the cost estimation process. Figure 1. Classical view of the algorithmic cost estimation process.  


An input requirement of an algorithmic model is to provide a metric to measure the size of the finished system. Typically lines of source code are used, this is obviously not known at the start of the project. SLOC is also very dependant on the programming language and programming environment, this is difficult to determine at an early stage in the problem especially as requirements are likely to be sketchy. Despite this SLOC has been the most widely used size metric in the past, but current trends indicate that it is fast becoming less stable. This is probably due to the changes in software development process in recent years highlighted with a tendency to use prototyping, case tools and so forth. An alternative is to use function points proposed by [ALBRECHT], which are related to the functionality of the software rather than its size. A more recent approach is to use object points. This is in comparison a new methodology and has not been publicised in the same depth as function points and SLOC. In essence the method is very similar to function points but counts objects instead of functions. Its recent rise has been prompted by the interest in the object orientation revolution.

Algorithmic models generally provide direct estimates of effort or duration. As shown in [figure 1](http://www.ecfc.u-net.com/cost/models.htm#Figure 1. Classical view of the algorithmic cost estimation process#Figure 1. Classical view of the algorithmic cost estimation process) the main input is usually a prediction of software size. Effort prediction models take the general form :

**effort = p\*S**

(1/productivity rate)

where p is a productivity constant and S is the size of the system.

Once the value for p is known. E.g. productivity = **450** source lines of code per month, making **p = 0.0022** and the size of the system has been estimated at 8500 KLOC.

 effort = 0.0022 \* 8500

**effort = 18.7 person moths**

 The example above assumes that the relationship between effort and size is a linear one. Most models allow for non-linear relationships by introducing economies or dis-economies of scale. The general formula being:

**effort = p \* Se**

A study published by [WALSTON & FELIX] which consisted of 60 projects at IBM federal systems division concluded that effort could be modelled as :

**effort = 5.2 KLOC 0.91**

This is an example of economies of scale. As the exponent 0.91 is less than 1. Walston and Felix observed the following results:

These findings indicate that there is greater productivity when building large software systems as opposed to small systems. However, the results can be justified as it is expected that larger teams can specialize and the overheads are of a relatively fixed size.A publication by [BANKER & KEMERER] discusses the existence of either economies and diseconomies of scale, for more information about this I strongly recommend reading this paper.

**Software Engineering Model Used**

The S/W Engineering Model used here is a combination of Spiral Model and Component Based Model because no single model can serve all the purpose for the development of the software, that’s why we are considering some features of “Spiral Model” and some features of “Component Based Model”.

The combinations of these two models form a “Hybrid Model”.

The Spiral Model

The spiral model, proposed by Boehm, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model. It provides the potential for rapid development of incremental versions of the software. Using the spiral model, software is developed in a series of incremental releases. During early iterations, the incremental release might be a paper model or prototype. During later iterations, increasingly more complete versions of the engineered system are produced.

A spiral model is divided into six task regions:

• Customer communication—tasks required to establish effective communication between developer and customer.

• Planning—tasks required to define resources, timelines, and other project related information.

• Risk analysis—tasks required to assess both technical and management risks.

• Engineering—tasks required to build one or more representations of the application.

• Construction and release—tasks required to construct, test, install, and provide user support and training).

• Customer evaluation—tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

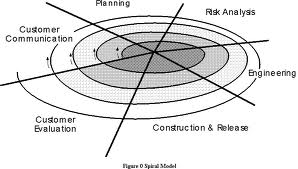
The spiral model is a realistic approach to the development of large-scale systems and software. Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level. The spiral model uses prototyping as a risk reduction mechanism but, more important, enables the developer to apply the prototyping approach at any stage in the evolution of the product. It maintains the systematic stepwise approach suggested by the classic life cycle but incorporates it into an iterative framework that more realistically reflects the real world. The spiral model demands a direct consideration of technical risks at all stages of the project and, if properly applied, should reduce risks before they become problematic.

Key Strength:-

* Efficiently applies risk driven approach.
* Tries to eliminate errors in early phases.
* Provides efficient mechanism for Quality Assurance.
* Works well fro a complex, dynamic and innovative projects.
* Re evaluation after each phase allows changes in user perspectives, technology advances or financial perspective.

Key Defects:-

* Lacks explicit process guidance to determine modular division, their objective, their constrains and alternatives.
* Due to continuous risk assessment at every stage it would become difficult to complete a project on a given deadline.



Description of Spiral model

Object-oriented technologies provide the technical framework for a component-based process model for software engineering. The object-oriented paradigm emphasizes the creation of classes that encapsulate both data and the algorithms used to manipulate the data. If properly designed and implemented, object-oriented classes are reusable across different applications and computer-based system architectures.

The component-based development model incorporates many of the characteristics of the spiral model hence can be easily use with Spiral Model. It is evolutionary in nature, demanding an iterative approach to the creation of software. However, the component-based development model composes applications from prepackaged software components called classes.

The engineering activity begins with the identification of candidate classes. This is accomplished by examining the data to be manipulated by the application and the algorithms that will be applied to accomplish the manipulation. Corresponding data and algorithms are packaged into a class.

Classes created in past software engineering projects are stored in a class library or repository. Once candidate classes are identified, the class library is searched to determine if these classes already exist. If they do, they are extracted from the library and reused. If a candidate class does not reside in the library, it is engineered using object-oriented methods. The first iteration of the application to be built is then composed, using classes extracted from the library and any new classes built to meet the unique needs of the application. Process flow then returns to the spiral and will ultimately re-enter the component assembly iteration during subsequent passes through the engineering activity.

Cost Estimation

It is the responsibility of the project manager to make accurate estimations of effort and cost. This is particularly true for projects subject to competitive bidding where a bid too high compared with competitors would result in loosing the contract or a bid too low could result in a loss to the organization. This does not mean that internal projects are unimportant. From a project leaders estimate the management often decide whether to proceed with the project. Industry has a need for accurate estimates of effort and size at a very early stage in a project. However, when software cost estimates are done early in the software development process the estimate can be based on wrong or incomplete requirements. A software cost estimate process is the set of techniques and procedures that an organization uses to arrive at an estimate. An important aspect of software projects is to know the cost; the major contributing factor is effort.

**Why SCE is difficult and error prone ?**

Software cost estimation requires a significant amount of effort to perform it correctly.

SCE is often done hurriedly, without an appreciation for the effort required.

You need experience at developing estimates, especially for large projects.

Human bias i.e An Estimator is likely to consider how long a certain portion of the system would take, and then to merely extrapolate this estimate to the rest of the system, ignoring the non-linear aspects of software development.

**The causes of poor and inaccurate estimation**

imprecise and drifting requirements

new software projects are nearly always different form the last.

software practitioners don't collect enough information about past projects.

estimates are forced to match the resources available.

**ALGORITHMIC COST MODELS:-**

To date most work carried out in the software cost estimation field has focused on algorithmic cost modeling. In this process costs are analyzed using mathematical formulae linking costs or inputs with metrics to produce an estimated output. The formulae used in a formal model arise from the analysis of historical data. The accuracy of the model can be improved by calibrating the model to your specific development environment, which basically involves adjusting the weightings of the metrics. There are a variety of different models available, the best known are Boehm's COCOMO [BOEHM-81], Putman's SLIM, and Albrecht's' function points [ALBR-83]. On an initial instinct you might expect formal models to be advantageous for their 'off-the-shelf' qualities, but after close observation this is regarded as a disadvantage by cost estimators due to the additional overhead of calibrating the system to the local circumstances. However, the more time spent calibrating a formal model the more accurate the cost estimate should be. A distinct disadvantage of formal models is the inconsistency of estimates, [KEMERER] conducted a study indicating that estimates varied from as much as 85 - 610 % between predicated and actual values. Calibration of the model can improve these figures, However, formal models still produce errors of 50-100%.

In terms of the estimation process , nearly all algorithmic models deviate from the classical view of the cost estimation process. Figure 1. Classical view of the algorithmic cost estimation process.  


An input requirement of an algorithmic model is to provide a metric to measure the size of the finished system. Typically lines of source code are used, this is obviously not known at the start of the project. SLOC is also very dependant on the programming language and programming environment, this is difficult to determine at an early stage in the problem especially as requirements are likely to be sketchy. Despite this SLOC has been the most widely used size metric in the past, but current trends indicate that it is fast becoming less stable. This is probably due to the changes in software development process in recent years highlighted with a tendency to use prototyping, case tools and so forth. An alternative is to use function points proposed by [ALBRECHT], which are related to the functionality of the software rather than its size. A more recent approach is to use object points. This is in comparison a new methodology and has not been publicised in the same depth as function points and SLOC. In essence the method is very similar to function points but counts objects instead of functions. Its recent rise has been prompted by the interest in the object orientation revolution.

Algorithmic models generally provide direct estimates of effort or duration. As shown in [figure 1](http://www.ecfc.u-net.com/cost/models.htm#Figure 1. Classical view of the algorithmic cost estimation process#Figure 1. Classical view of the algorithmic cost estimation process) the main input is usually a prediction of software size. Effort prediction models take the general form :

**effort = p\*S**

(1/productivity rate)

where p is a productivity constant and S is the size of the system.

Once the value for p is known. E.g. productivity = **450** source lines of code per month, making **p = 0.0022** and the size of the system has been estimated at 8500 KLOC.

 effort = 0.0022 \* 8500

**effort = 18.7 person moths**

 The example above assumes that the relationship between effort and size is a linear one. Most models allow for non-linear relationships by introducing economies or dis-economies of scale. The general formula being:

**effort = p \* Se**

A study published by [WALSTON & FELIX] which consisted of 60 projects at IBM federal systems division concluded that effort could be modelled as :

**effort = 5.2 KLOC 0.91**

This is an example of economies of scale. As the exponent 0.91 is less than 1. Walston and Felix observed the following results:

These findings indicate that there is greater productivity when building large software systems as opposed to small systems. However, the results can be justified as it is expected that larger teams can specialize and the overheads are of a relatively fixed size.A publication by [BANKER & KEMERER] discusses the existence of either economies and diseconomies of scale, for more information about this I strongly recommend reading this paper.

GANTT CHART

A grant chart or bar chart is the simplest form of formal project management. The Gantt chart is used almost exclusively for scheduling purpose and therefore controls only the time dimension.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 10jan  2013 | 13jan  2013 | 15jan  2013 | 23jan  2013 | 4feb  2013 | 8feb  2013 |
| Start | Information Gathering |  |  |  |  |  |
|  |  | Problem Identification |  |  |  |  |
|  |  |  | Requirement  Analysis |  |  |  |
|  |  |  |  | Risk Analysis |  |  |
|  |  |  |  |  | Cost Analysis |  |
|  |  |  |  |  |  | Prototype  Designing |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15feb  2013 | 19feb  2013 | 25feb  2013 | 5mar  2013 | 10mar  2013 | 14mar  2013 | 15mar  2013 |
| Deployment  &validation |  |  |  |  |  |  |
|  | Modules |  |  |  |  |  |
|  |  | System  Integration |  |  |  |  |
|  |  |  | Testing |  |  |  |
|  |  |  |  | Deployment |  |  |
|  |  |  |  |  | Documentation | Finish |

# **Pert Chart**

# Program Evaluation and Review Technique

T=5d S2 T=12d T=20d S5

S1

S4 T=1md T=7d T=10d T=1md

S3 S6

T=15d

S7

S8 T=3d

Project Planning

1. Project Initiation

The Initiation Phase is the first phase in the project. In this phase a business problem (or opportunity) is identified and a business case which provides various solution options is defined. A feasibility study is then conducted to investigate the likelihood of each solution option addressing the business problem and a final recommended solution is put forward. Once the recommended solution is approved, a project is initiated to deliver the approved solution. A ‘Terms of Reference’ is completed, which outlines the objectives, scope and structure of the new project, and a Project Manager is appointed. The Project Manager begins recruiting a project team and establishes a Project Office environment. Approval is then sought to move into the detailed planning phase.

2.Project Planning

Once the scope of the project has been defined in the terms of reference, the project enters the detailed planning phase. This involves the creation of a:

* Project plan (outlining the activities, tasks, dependencies and timeframes).
* Resource Plan (listing the labor, equipment and materials required)
* Financial Plan (identifying the labor, equipment and materials costs)
* Quality Plan (providing quality targets, assurance and control measures)
* Risk Plan (highlighting potential risks and actions taken to mitigate them)
* Acceptance Plan (listing the criteria to be met to gain customer acceptance)
* Communication Plan (listing the information needed to inform stakeholders)
* Procurement Plan (identifying products to be sourced from external suppliers).
* At this point the project has been planned in detail and is ready to be executed.

3.Project Execution

This phase involves the execution of each activity and task listed in the Project plan. While the activities and tasks are being executed, a series of management processes are undertaken to monitor and control the deliverables being output by the project. This includes the identification of changes, risks and issues, the review of deliverable being produced against the acceptance criteria. Once all of the deliverables have been produced and the customer has accepted the final solution, the project is ready for closure.

4.Project Closure

Project Closure involves releasing the final deliverables to the customer, handling over project documentation, terminating supplier contracts, releasing project resources and communicating the closure of the project to all stakeholders. The Last remaining step is to undertake a Post implementation Review to quantify the overall success of the project and list any lessons learnt for future projects.

The following sections provide a more detailed description of each phase and list document templates which provide the project Manger with guidance on how to complete each phase successfully.

Initiation

The initiation phase essentially involves the project ‘start-up’. It is the phase with in which the business problem or opportunity is identified, the solution is agreed, a project formed to produce the solution and a project team appointed. The diagram below depicts the activities undertaken:

Develop Business Case

Once a business problem or opportunity has been identified, a business case is prepared. This includes:

* A detailed definition of the problem or opportunity
* A analysis of the potential solution options available. For each options, the potential benefits, costs, risks and issues are documented. A formal feasibility study may be commissioned if the feasibility of any particular solution option is not clear.
* The recommended solution and a generic implementation plan.

The business case is approved by the project sponsor and the required funding is allocated o proceed with the project.

Perform Feasibility Study

At any stage during (or after) the development of a business case, a formal feasibility study may be commissioned. The purpose is to assess the likelihood of a particular solution option’s achieving the benefits outlined in the business case. The feasibility study will also investigate whether the forecast costs are reasonable, the solution is achievable, the risks rare acceptable and/or any likely issues are avoidable.

Establish Terms of references

After the solution has been agreed and funding Allocated, a project is formed. The Terms of Reference defined the vision, objectives, scope the organization structure (roles and responsibilities) and a summarized plan of the activities, resources and funding require to undertake the project. Finally, any risks, issues, planning assumptions and constraints are listed.

MODULARISATION DETAILS

The software will be designed according to the various modules required to fulfill all the requirements uncovered in our requirement analysis. The whole system can be divided into a number of modules. The following modules can be recognized in our project:

There are 7 modules in the project. They are as follows

1.LOGIN MODULE

Login module is the first and the most important module of the project “Shopping Hub – An Online Store”. It plays very important and vital role in the security. As only the authorized user can access it and can do the shopping according to it no other user can access it as it required a user name and password which is used to access the software.

2.ADMINISTRATOR MODULE

This module is used for administrative properties like add, delete, modify etc... Any changes to be made in the product list are done by the help of this module. This module is used by the administrator no one has right to access this module.

3.CUSTOMER MODULE

This module is used for the registration of the customer which holds each and every detail of the customer so that if any fraud case is done by the customer, customer can be easily traced. It keeps the details like name, address, phone number, e-mail address etc.

4.SEARCH MODULE

The very important module of the software as the basis of the software is searching, the product which the customer demands is been searched by the help of this module. This module displays all the items which are presently present for the sale to the customer.

5.ORDER MODULE

This module is used efficiently by the customer as well as by the administrator as customer places the order by the help of this module only and administrator checks the order to be delivered by the help of this module only.

6.PAYMENT MODULE

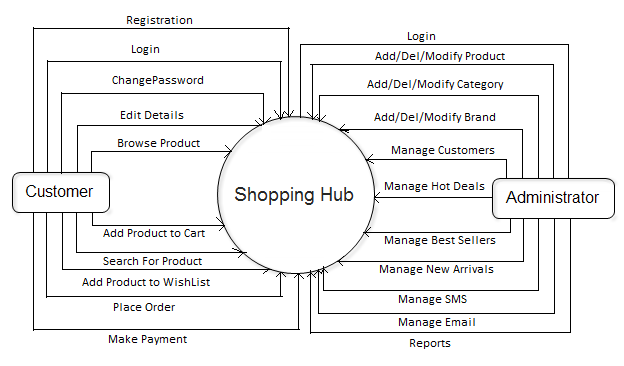
This module has its own role in the software it is used when the customer had selected the item which he/she has to purchase. This module access the credit card sections for the online payment of the purchase which in turn generate the payment slip which can be printed through the printer very easily.

7.CART MODULE

This is a crucial module which adds various items to the cart then it asks for a confirmation of added items, here the customer can add or delete the items according to his needs and then he is asked for the payment.

DATA FLOW DIAGRAMS

LEVEL 0 (CONTEXT LEVEL)



LEVEL 1 (CUSTOMER)

tbl\_UserAccount

Browse

Or

Search

tbl\_Product

Registration

tbl\_UserAccount

Add

Login

Customer

tbl\_ShoppingCart

Add

tbl\_WishList

Updates

Makes

Payment

tbl\_Order

tbl\_UserProfile

LEVEL 1 (ADMINISTRATOR)

tbl\_HotDeal

tbl\_Brands

tbl\_UserAccount

tbl\_Products

tbl\_Category

tbl\_BestSeller

Administrator

tbl\_NewArrival

tbl\_Stock

tbl\_Admin

tbl\_Suppliers

tbl\_SMS

tbl\_Email

LEVEL 2 (CUSTOMER)

Registartion

tbl\_UserAccount

Details Added

Add Details

User Registered

Enter User Details

Authentication

tbl\_UserAccount

Email &

Password

Verified

Login Succeed

Login Details

Forgot Password

tbl\_UserAccount

Password Reset

Email,

Security Question

& Answer

Password Reset Request

Information

Search/Browse Product

tbl\_Product

Customer

tbl\_Category

tbl\_Brand

Add Product to Cart

Customer

tbl\_ShoppingCart

tbl\_UserAccount

Add Product to Wish List

Customer

tbl\_WishList

tbl\_UserAccount

Place Order & Make Payment

View Selected Item

Add Item

Delete Item

Modify Item

Card Detail

tbl\_Order

Bank Detail

LEVEL 2 (ADMINISTRATOR)

Admin Login

tbl\_Admin\_

UserName

&

Password

Verified

Login Details

Login Succeed

Manage Product/Brand/Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

tbl\_Product

tbl\_Brand

tbl\_Category

Manage Hot Deals/Bestseller/New Arrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

tbl\_HotDeals

tbl\_Bestseller

tbl\_NewArrival

Manage Inventory

tbl\_Sales

tbl\_Stock

tbl\_Purchase

Manage SMS/E-Mail

tbl\_SMS

tbl\_Email

Reports Generation

tbl\_Purchase

tbl\_Purchase

tbl\_Purchase

ER DIAGRAM

has

buys

manage

CUSTOMER

PRODUCT

has

added to

makes

made

for

CART

PAYMENT

CATEGORY

1

M

M

M

BRAND

1

ADMINISTRATOR

1

PURCHASE

manage

SALES

maintains

STOCK

M

DATABASE DESIGN

tbl\_Category (PK:Cat\_ID)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Cat\_id | Integer |
| Name | Nvarchar(25) |
| Description | Nvarchar(100) |
| Image | Varbinary |
| Active | Nvarchar(5) |

tbl\_product (PK:Prod\_id FK:Cat\_id Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Prod\_id | Integer |
| Name | Nvarchar(25) |
| Purchase\_price | Integer |
| Sale\_price | Integer |
| Brand\_id | Integer |
| Cat\_id | Integer |
| Description | Nvarchar(100) |
| Image | Varbinary |
| Rating | Integer |
| Active | Nvarchar(5) |

tbl\_brand (PK:Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Brand\_id | Integer |
| Name | Nvarchar(25) |
| Logo | Varbinary |
| Description | Nvarchar(100) |
| Active | Nvarchar(5) |

tbl\_wishlist (PK:Wl\_id FK:Cust\_id,Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Wl\_id | Integer |
| Cust\_id | Integer |
| Name | Nvarchar(25) |
| Prod\_id | Integer |
| Insert\_date | Datetime |

tbl\_Admin (PK:Login\_name)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Login\_name | Nvarchar(30) |
| Password | Nvarchar(30) |

tbl\_Order (PK:Order\_id FK:Cust\_id )

|  |  |
| --- | --- |
| Column Name | Data Type |
| Order\_id | Integer |
| Cust\_id | Integer |
| Order\_date | Datetime |
| Order\_amt | Integer |
| Discount | Integer |
| Shipping\_amt | Integer |
| Tax\_amt | Integer |
| Net\_amt | Integer |
| Shipping\_date | Datetime |
| Shipping\_address | Nvarchar(150) |
| Billing\_address | Nvarchar(150) |
| Status | Nvarchar(15) |

tbl\_ShoppingCart (PK:Cart\_id FK:Cust\_id FK:Product\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Cart\_id | Integer |
| Cust\_id | Integer |
| Session\_id | Integer |
| Product\_id | Integer |
| Quantity | Integer |
| Price | Integer |

tbl\_UserAccount (PK:Uid)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Uid | Integer |
| Username | Nvarchar(25) |
| Password | Nvarchar(25) |
| Hint\_Question | Nvarchar(50) |
| Answer | Nvarchar(30) |
| Status | Nvarchar(5) |

tbl\_UserProfile (PK:Uid)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Uid | Integer |
| Fname | Nvarchar(25) |
| Lname | Nvarchar(25) |
| Dob | Datetime |
| Mobile\_no | Nvarchar(10) |
| Gender | Nvarchar(6) |
| Email | Nvarchar(50) |
| City | Nvarchar(15) |
| State | Nvarchar(15) |
| Country | Nvarchar(15) |
| Aboutme | Nvarchar(100) |
| Pincode | Nvarchar(6) |

tbl\_HotDeals (PK:Hd\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Hd\_id | Integer |
| Prod\_id | Integer |
| Offer\_Price | Integer |
| Description | Nvarchar(100) |
| Active | Nvarchar(5) |
| Image | Varbinary |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_NewArrival (PK:Na\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Na\_id | Integer |
| Prod\_id | Integer |
| Price | Integer |
| Active | Nvarchar(5) |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_BestSeller (PK:Bs\_id FK:Prod\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Bs\_id | Integer |
| Prod\_id | Integer |
| Price | Integer |
| Active | Nvarchar(5) |
| Start\_date | Datetime |
| End\_date | Datetime |

tbl\_Sms (PK:Sms\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Sms\_id | Integer |
| Type | Nvarchar(10) |
| Mobile\_no | Nvarchar(10) |
| Text | Nvarchar(160) |
| Sendtime | datetime |

tbl\_Email (PK:E\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| E\_id | Integer |
| Type | Nvarchar(10) |
| Email\_id | Nvarchar(50) |
| Subject | Nvarchar(25) |
| Message | Nvarchar(250) |
| Send\_time | datetime |

tbl\_Stock (PK:Stock\_id FK:Prod\_id Brand\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Stock\_id | Integer |
| Prod\_id | Integer |
| Brand\_id | Integer |
| Prod\_name | Nvarchar(25) |
| Description | Nvarchar(150) |
| Quantity | Integer |
| Status | Nvarchar(5) |

tbl\_Supplier (PK:Supp\_id)

|  |  |
| --- | --- |
| Column Name | Data Type |
| Supp\_id | Integer |
| Supp\_name | Nvarchar(25) |
| Address | Nvarchar(60) |
| Prod\_sales | Nvarchar(20) |
| Email\_id | Nvarchar(50) |
| Mobile\_no | Nvarchar(10) |
| TIN\_no | Integer |

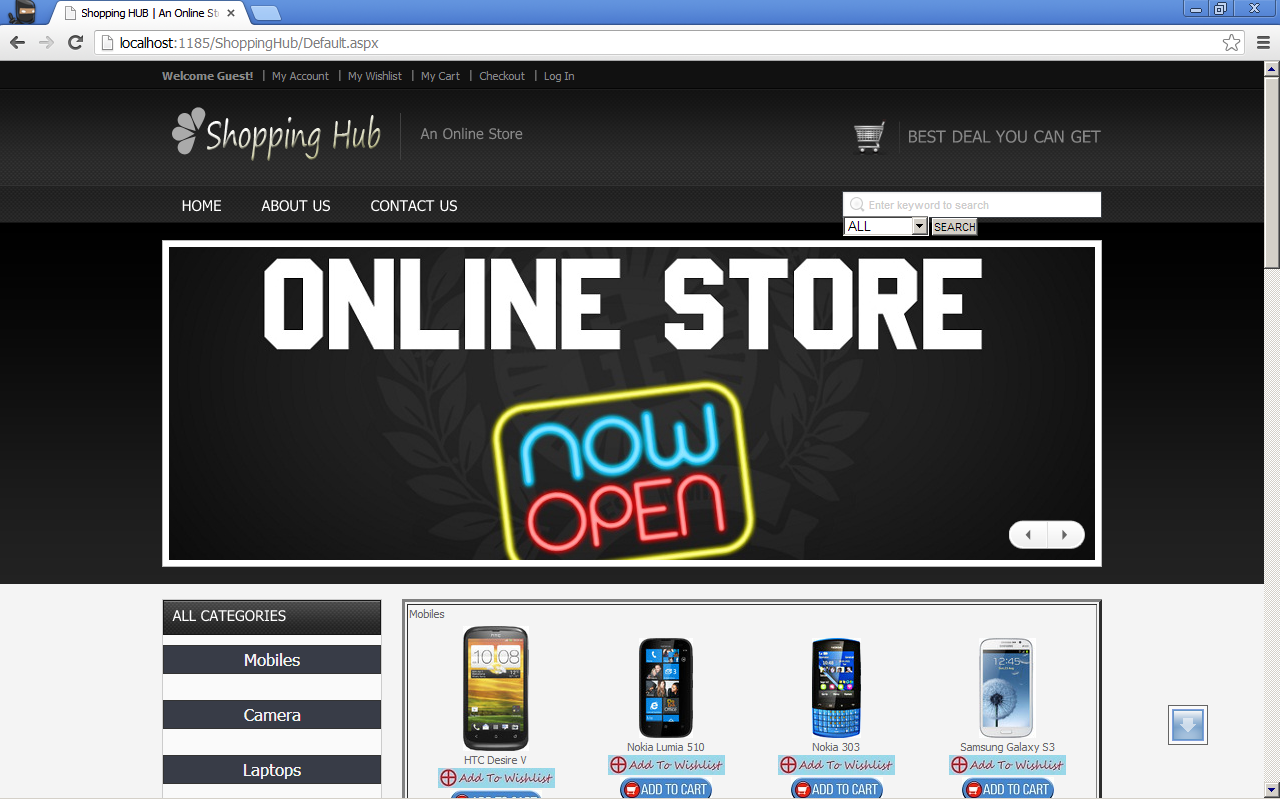
tbl\_Sales

|  |  |
| --- | --- |
| Column Name | Data Type |
| Sales\_id | Integer |
| Prod\_id | Integer |
| Cust\_id | Integer |
| Date\_sold | Datetime |
| Qty\_sold | Integer |

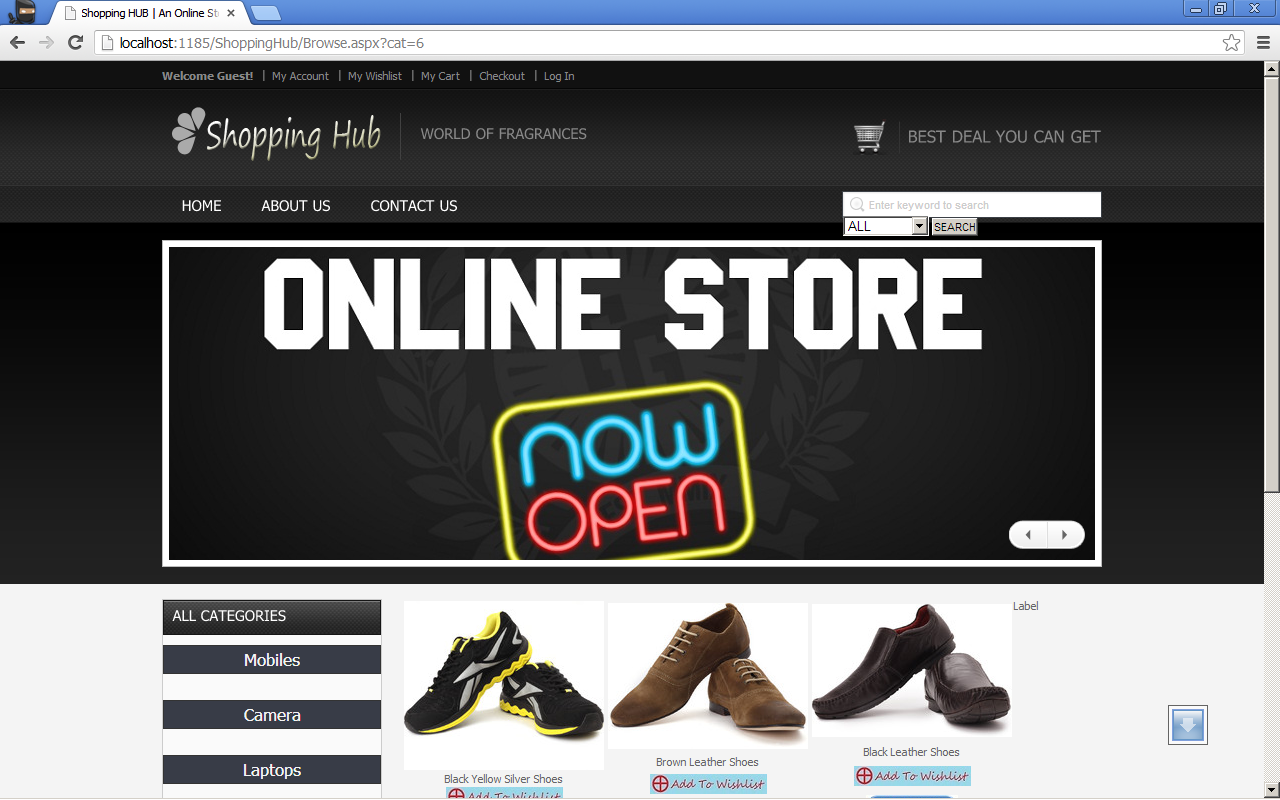
tbl\_Purchase (PK:Pur\_id FK:Supp\_id Prod\_id)

|  |  |
| --- | --- |
| Category | Datatype |
| Pur\_id | Integer |
| Supp\_id | Integer |
| Prod\_id | Integer |
| Quantity | Integer |
| Date | datetime |

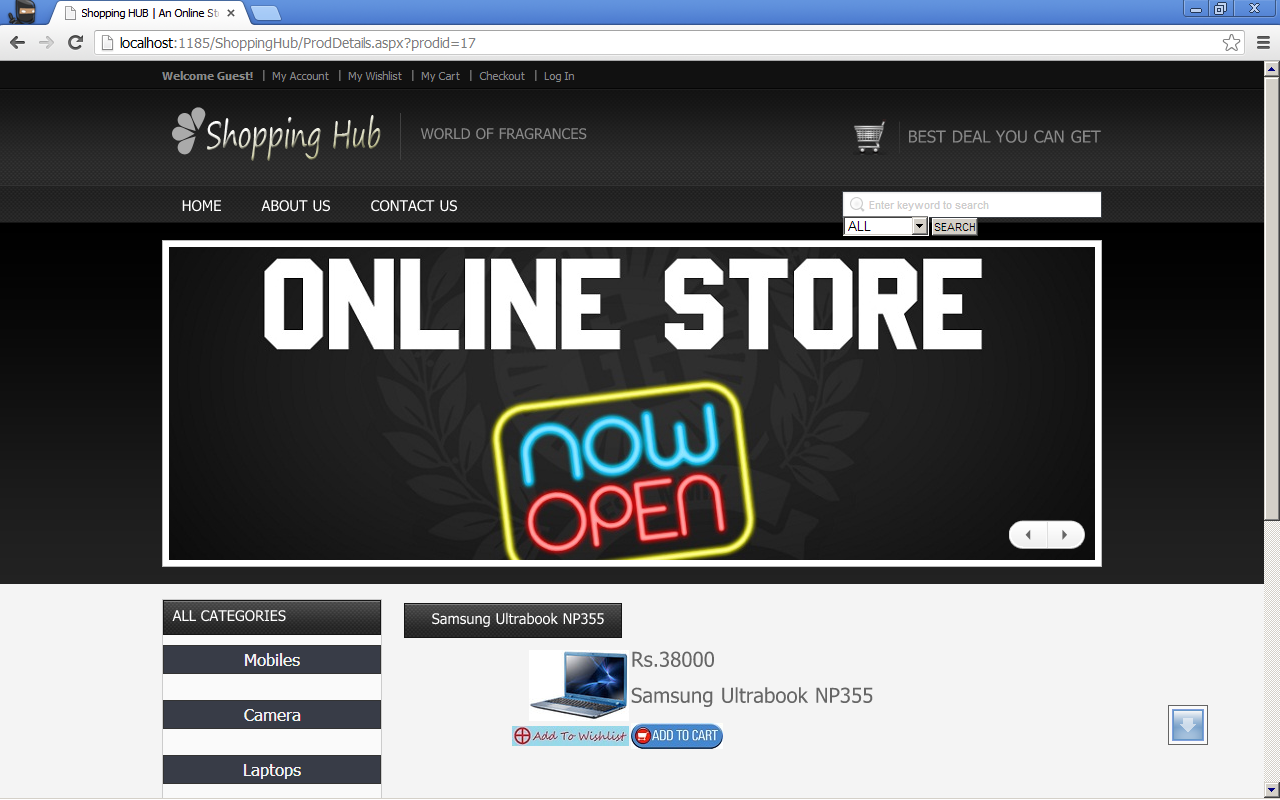
Default.aspx



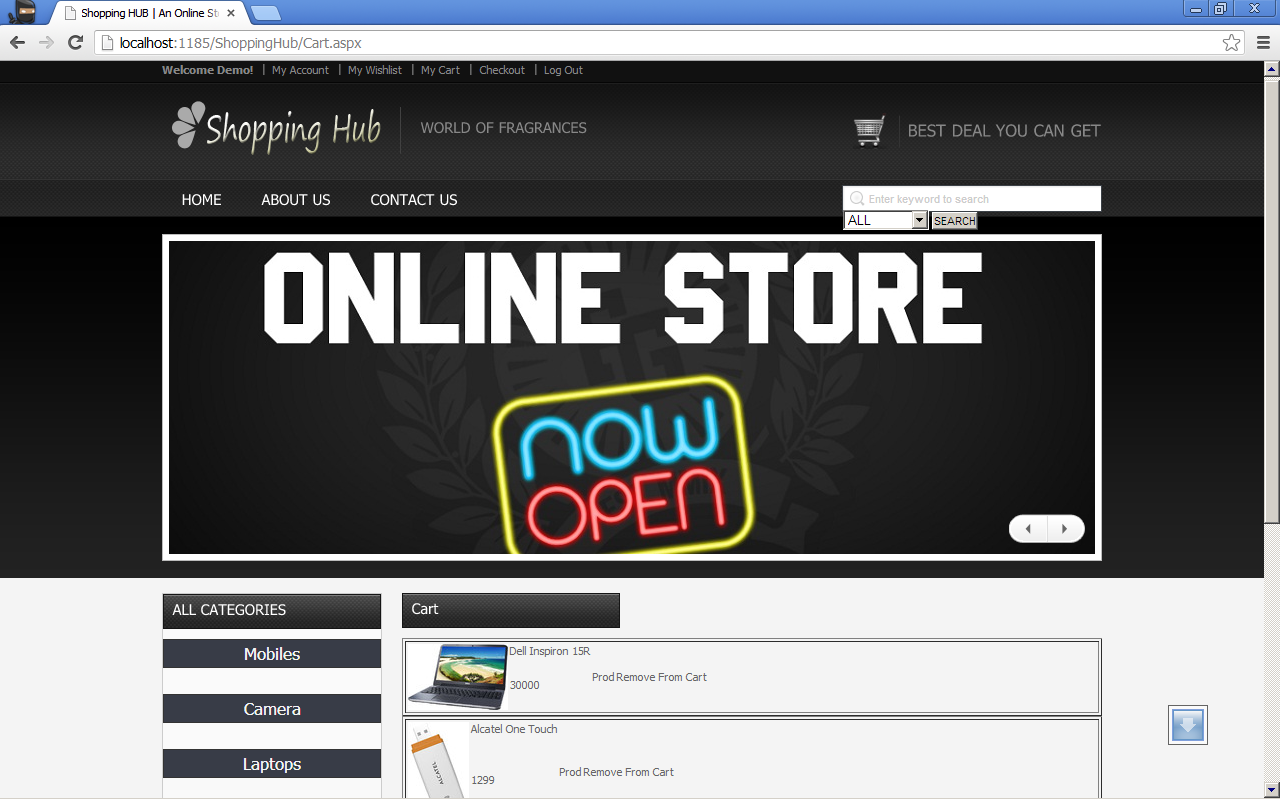
Browse.aspx



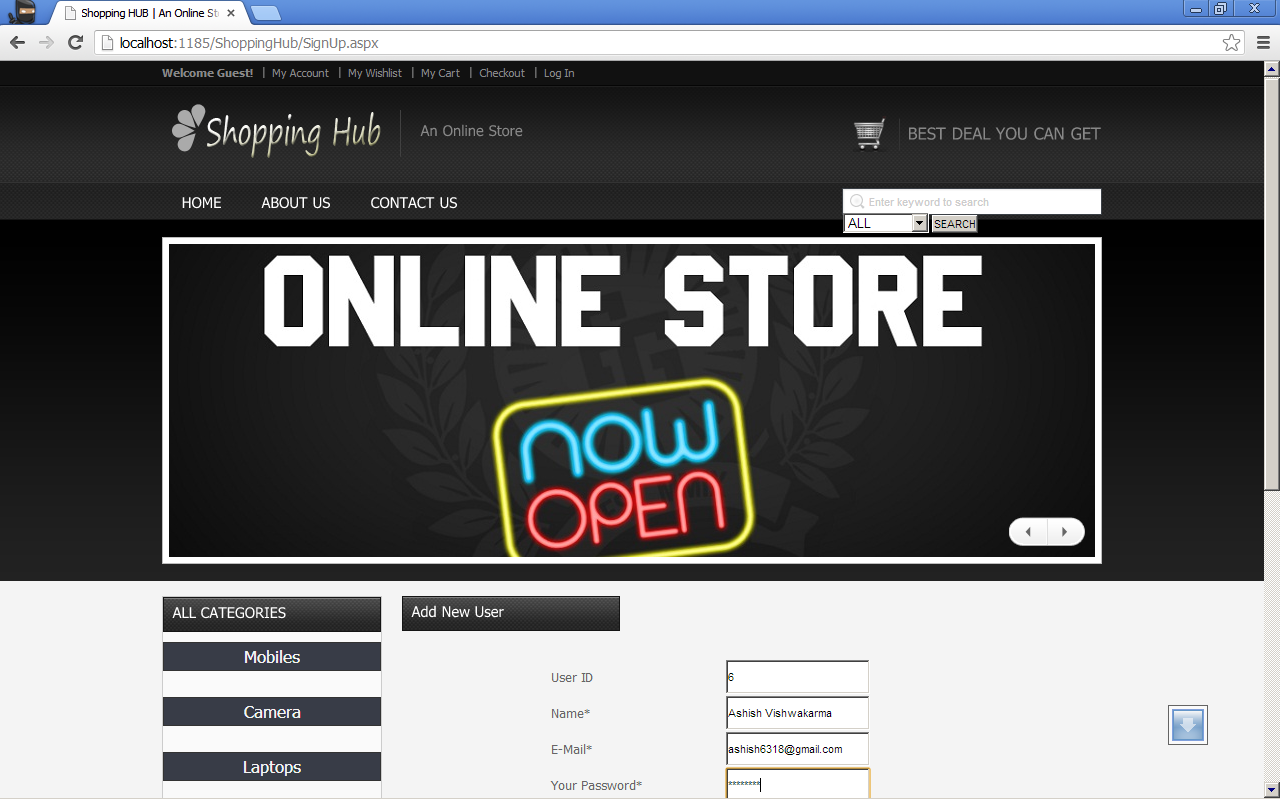
ProdDetails.aspx



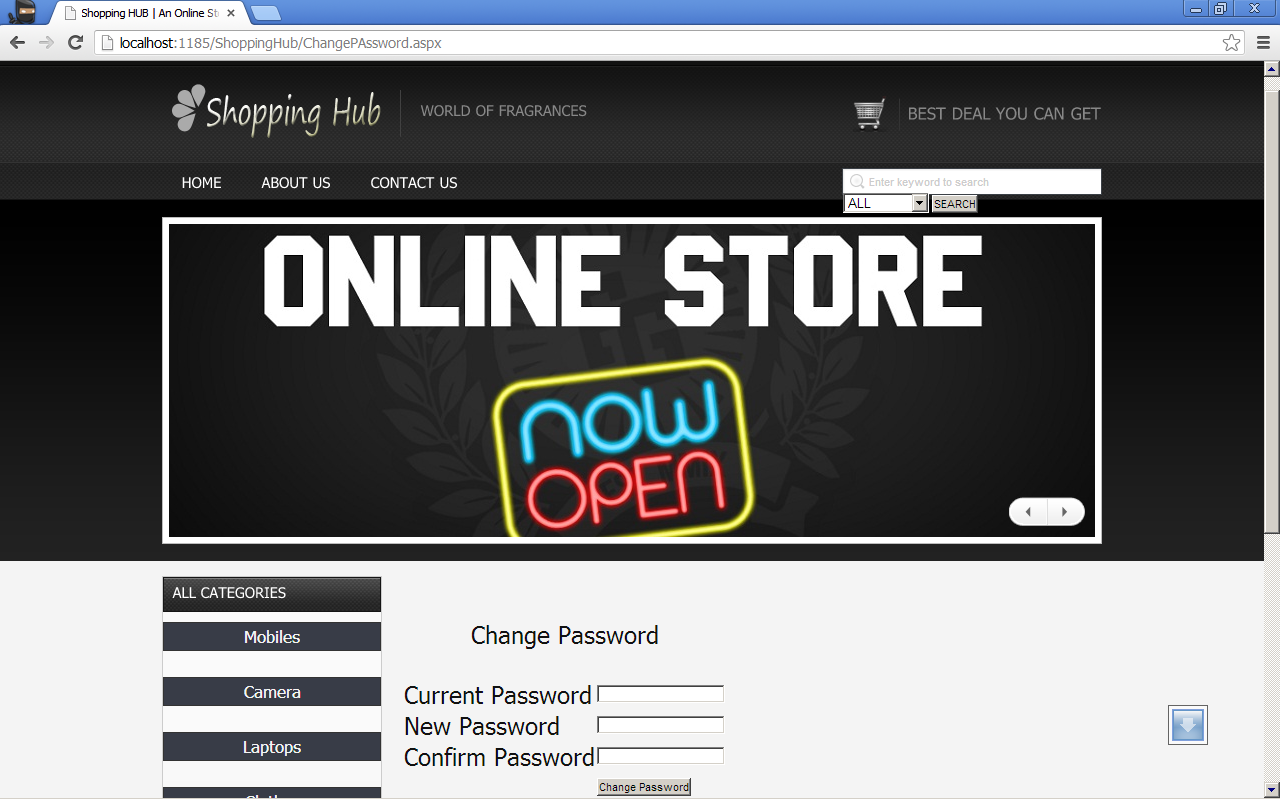
Cart.aspx



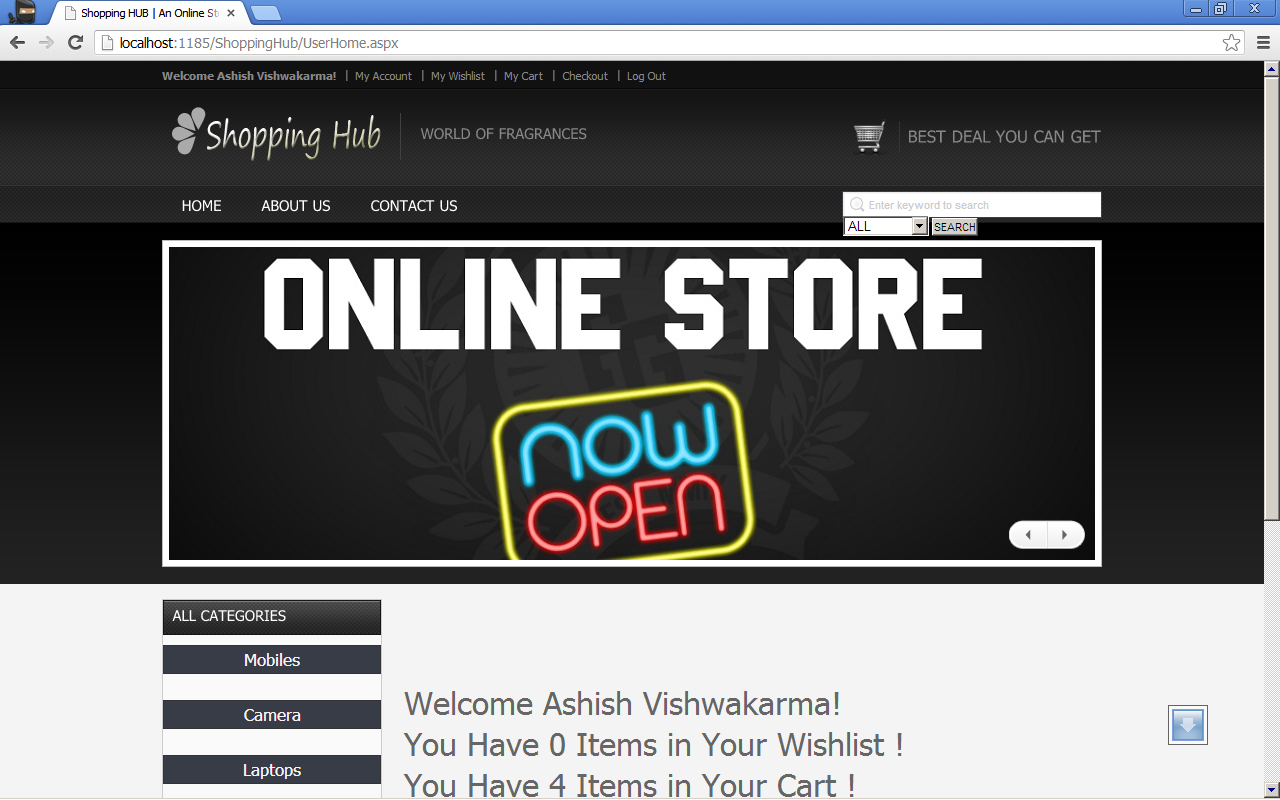
SignUp.aspx



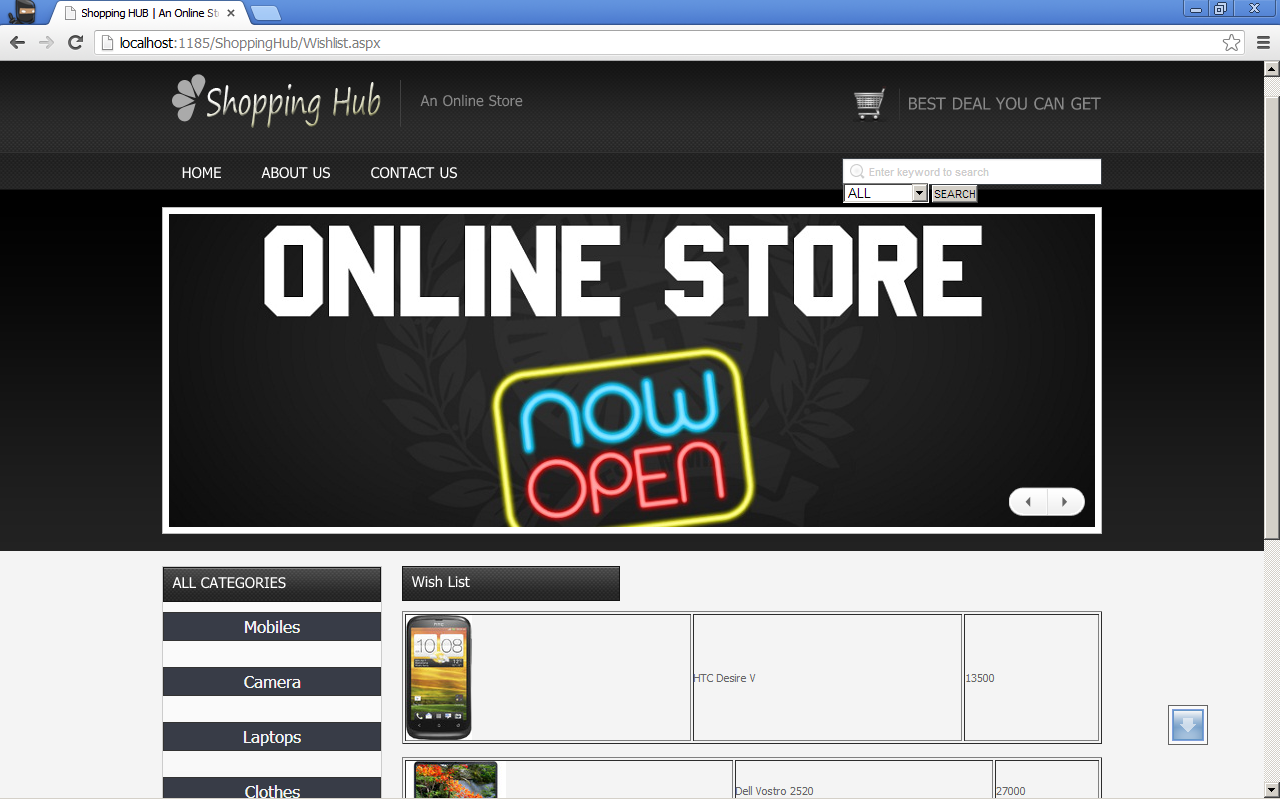
ChangePassword.aspx



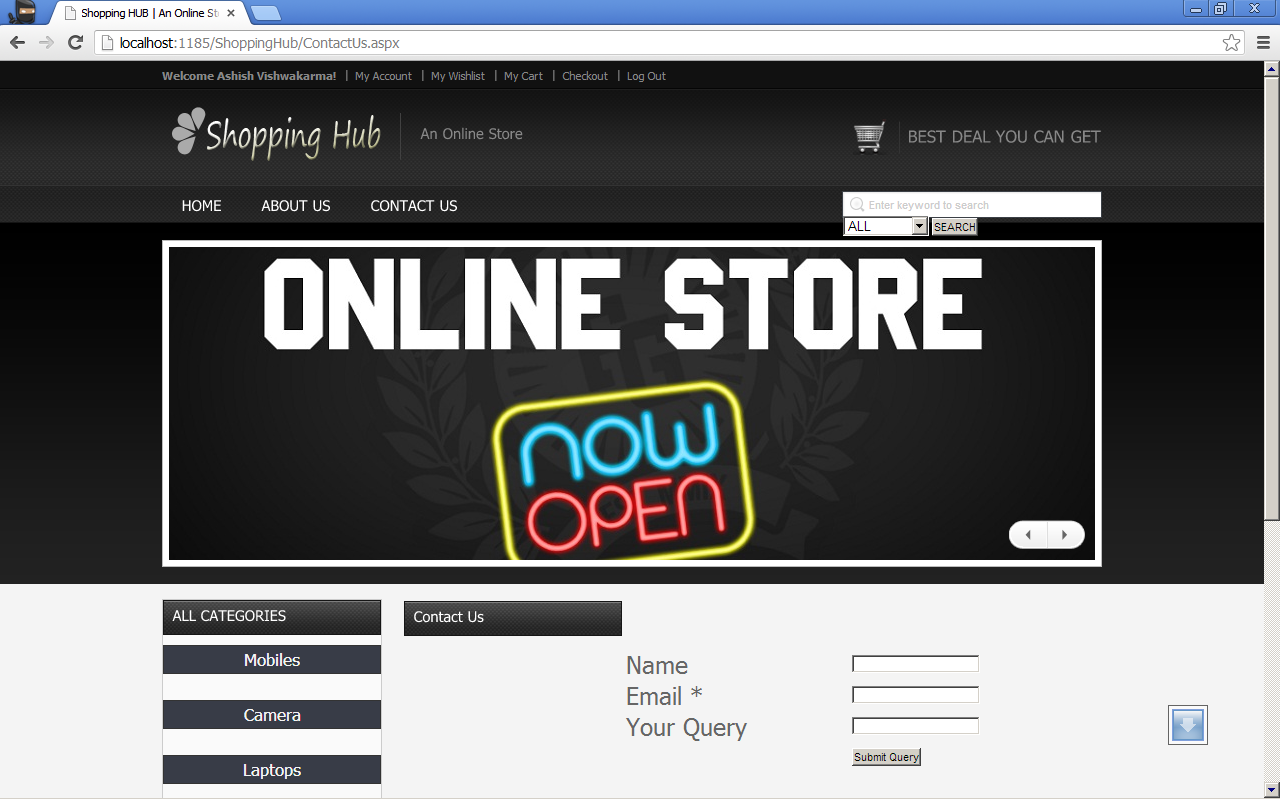
UserHome.aspx



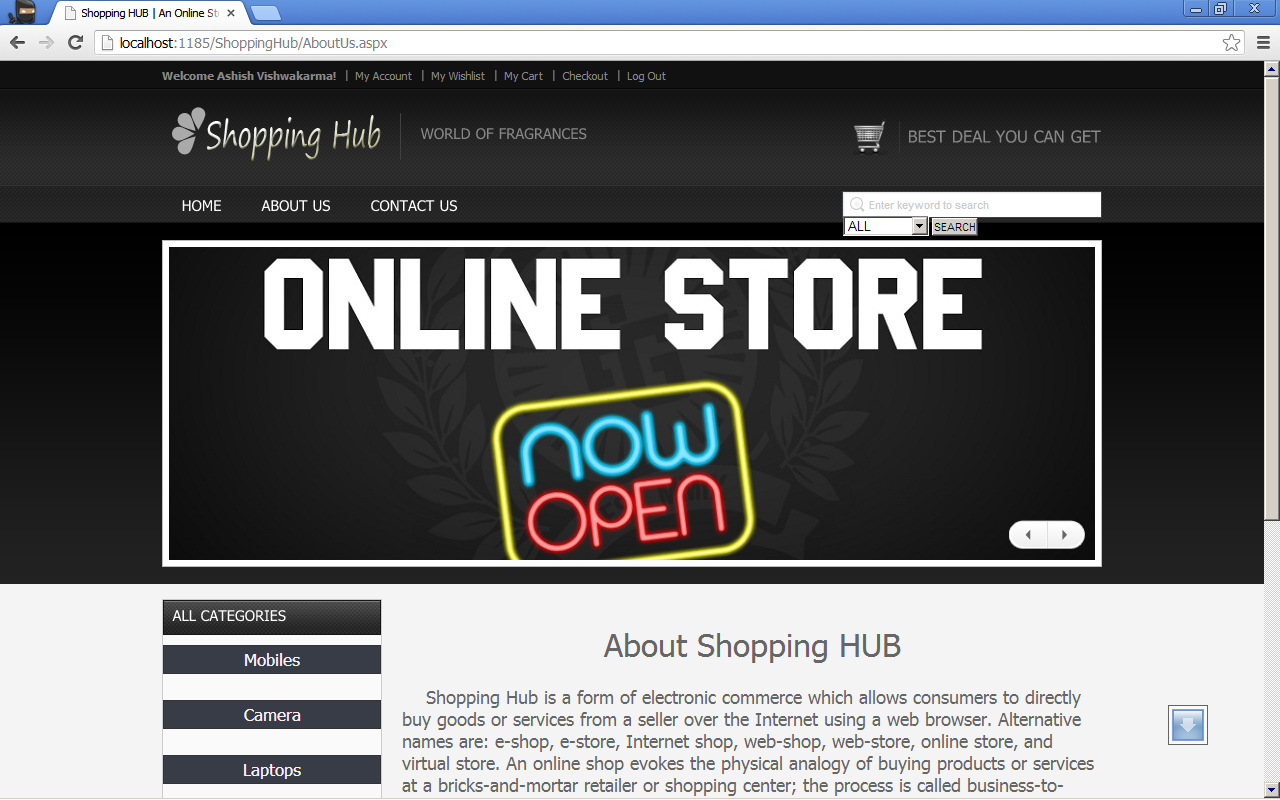
WishList.aspx



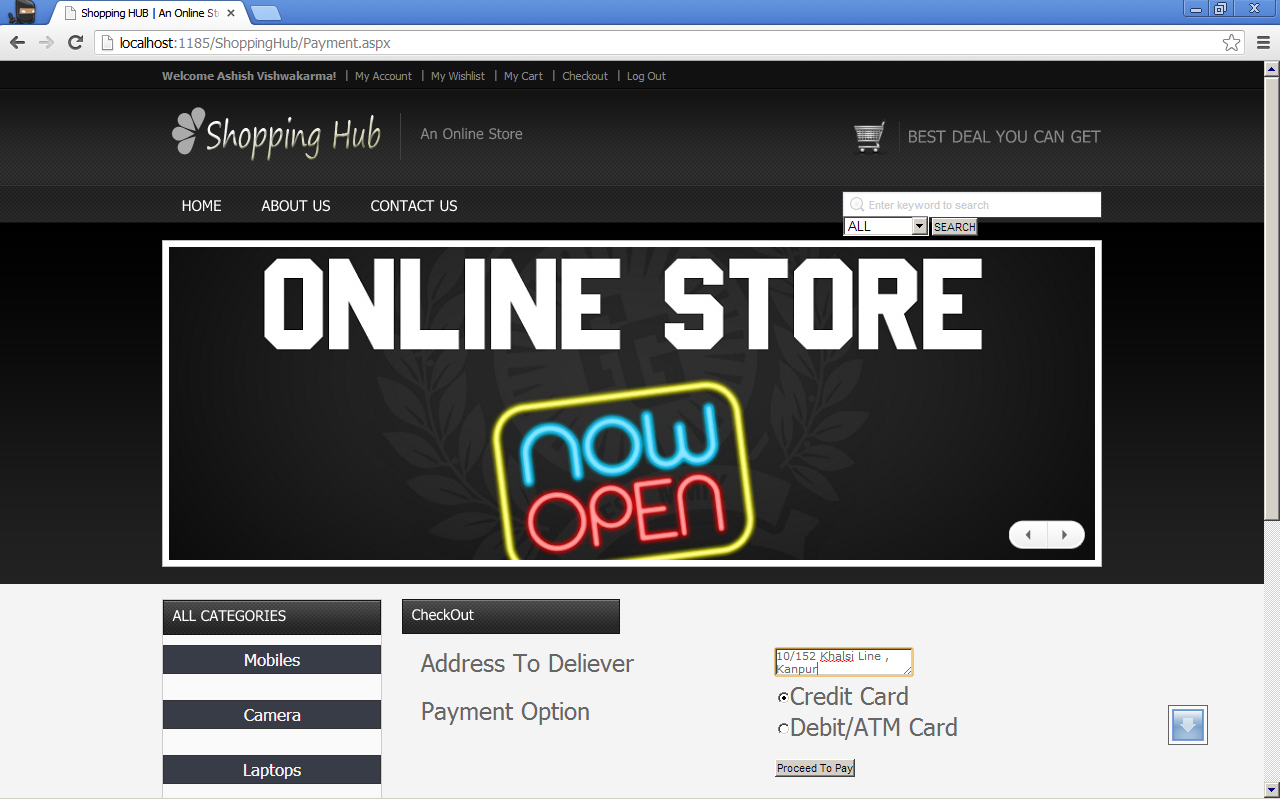
ContactUs.aspx



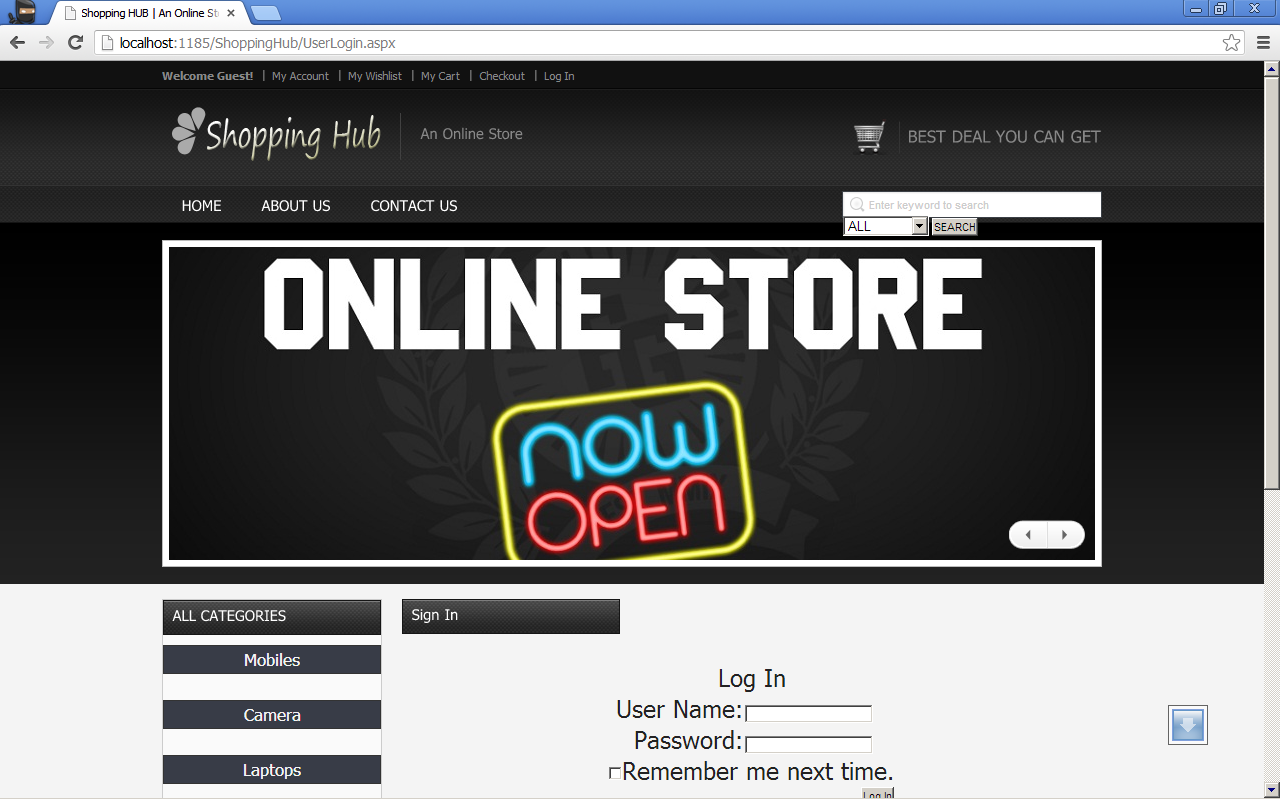
AboutUs.aspx



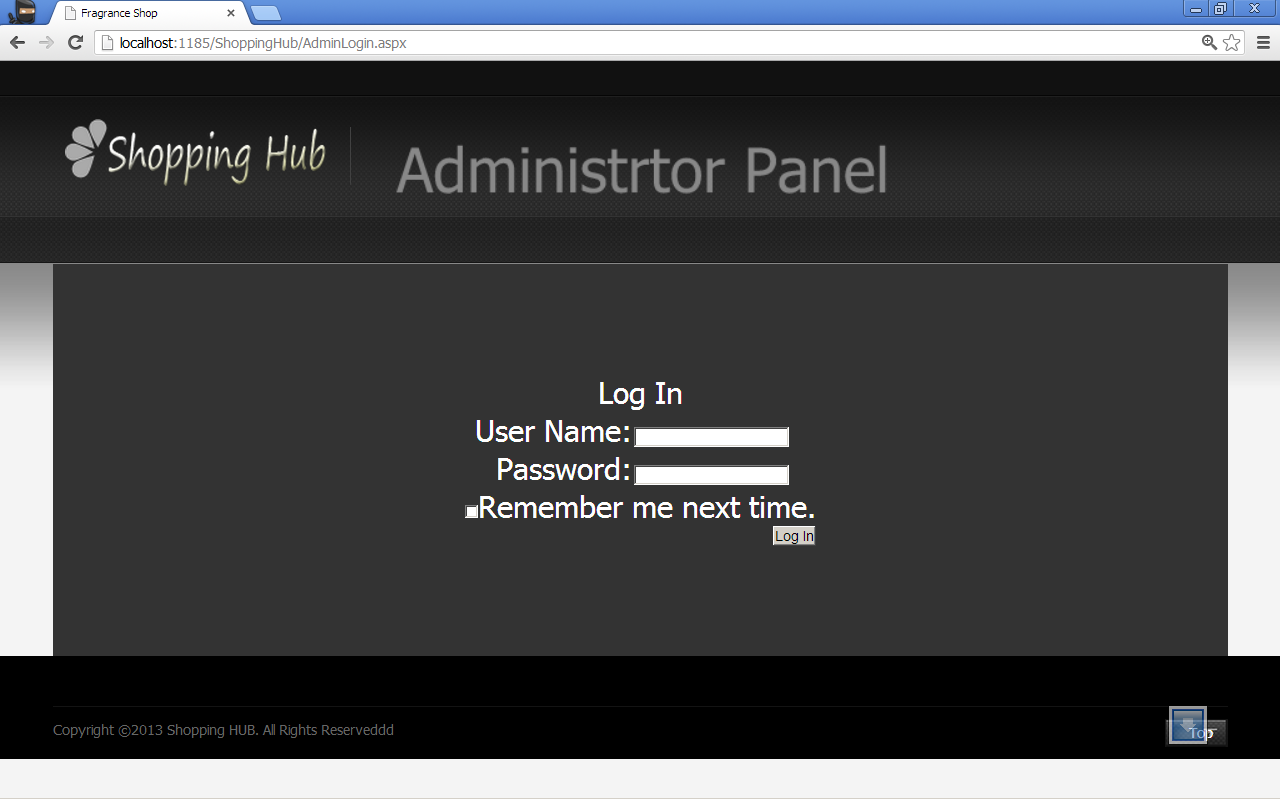
Payment.aspx



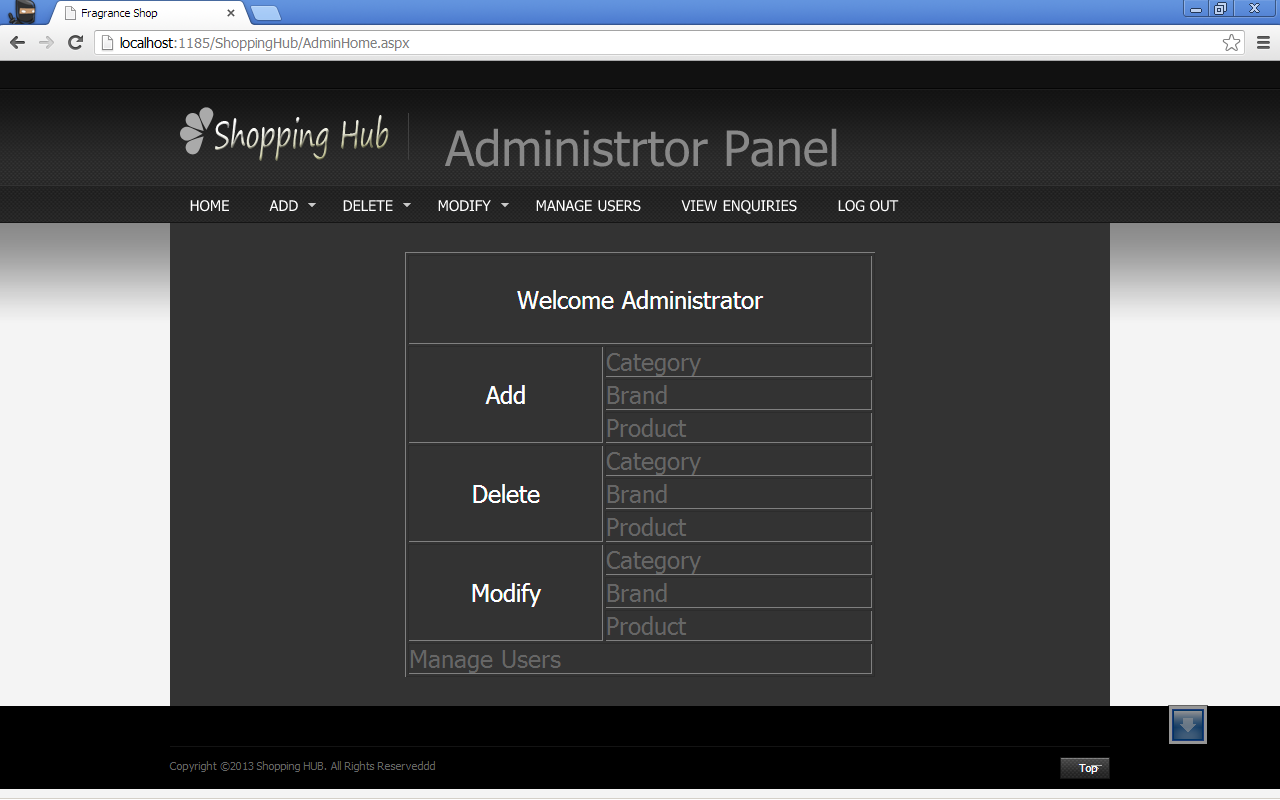
UserLogin.aspx



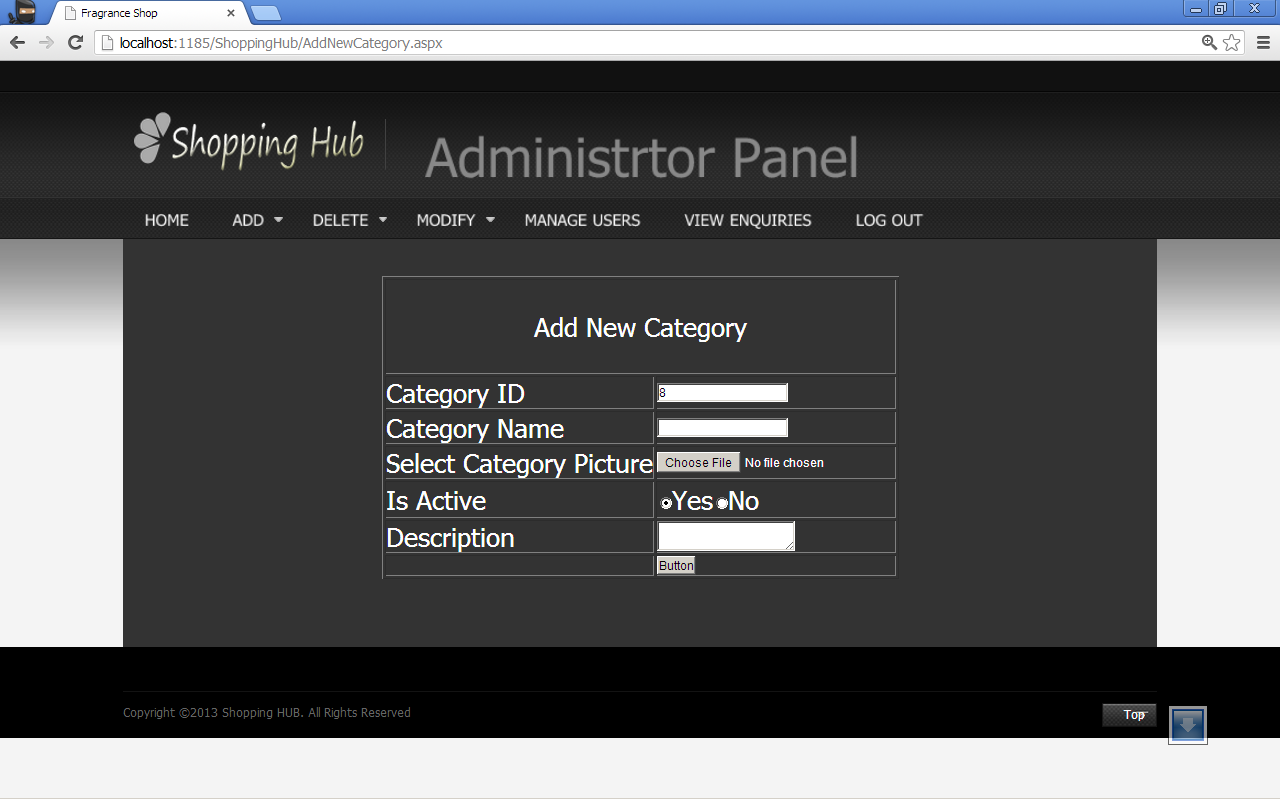
AdminLogin.aspx



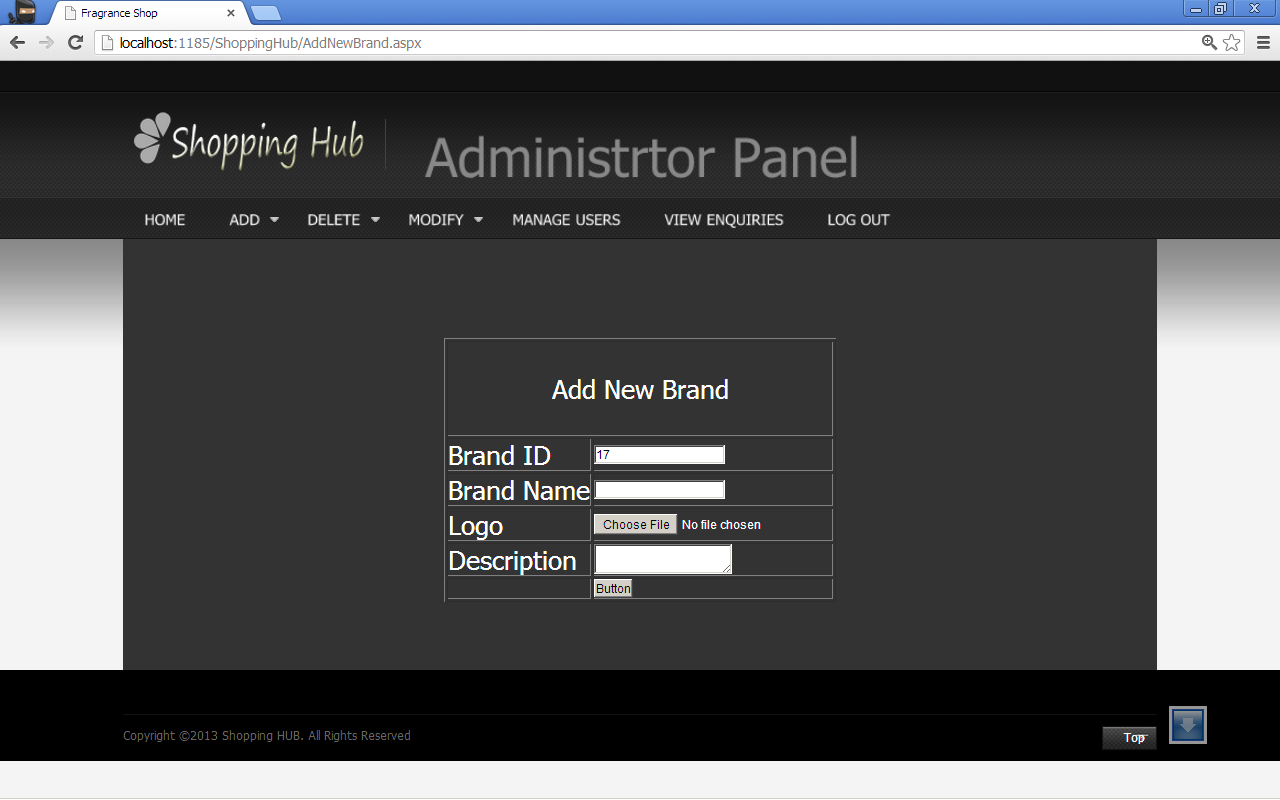
AdminHome.aspx



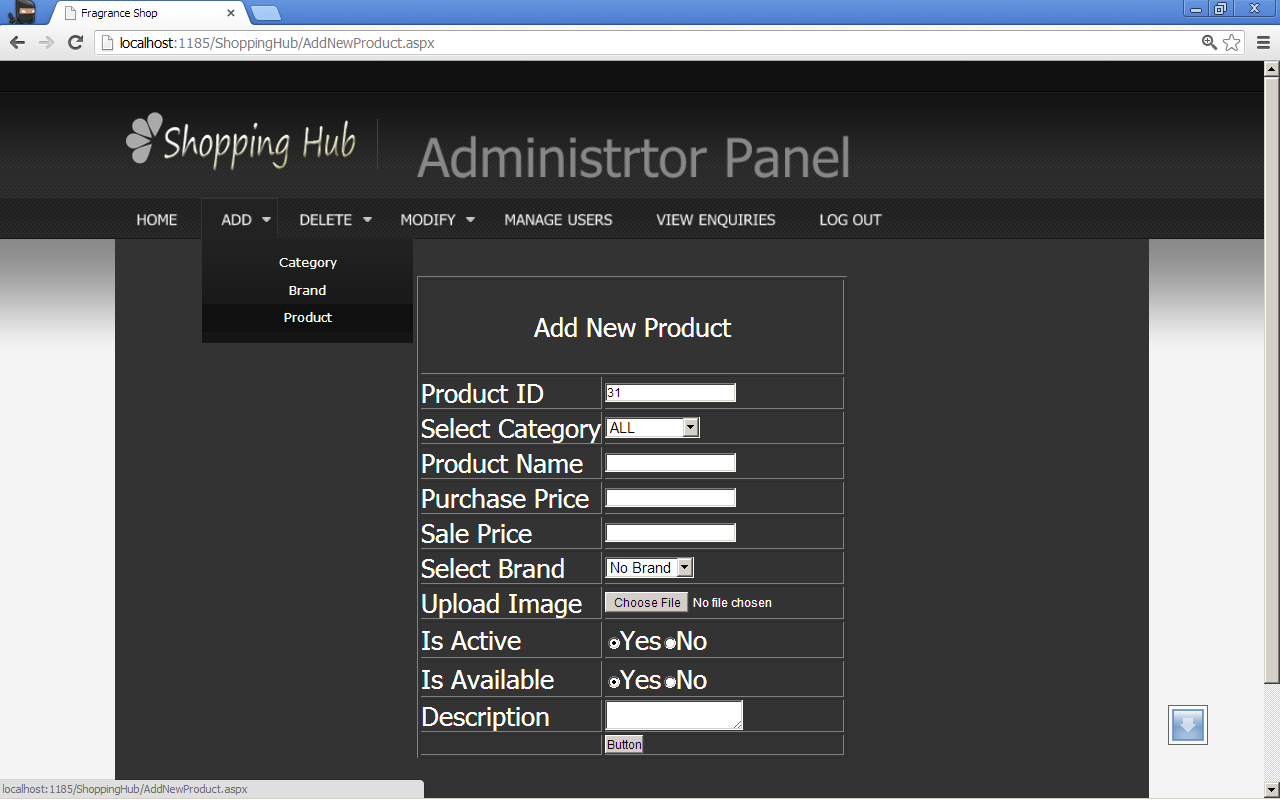
AddNewCategory.aspx



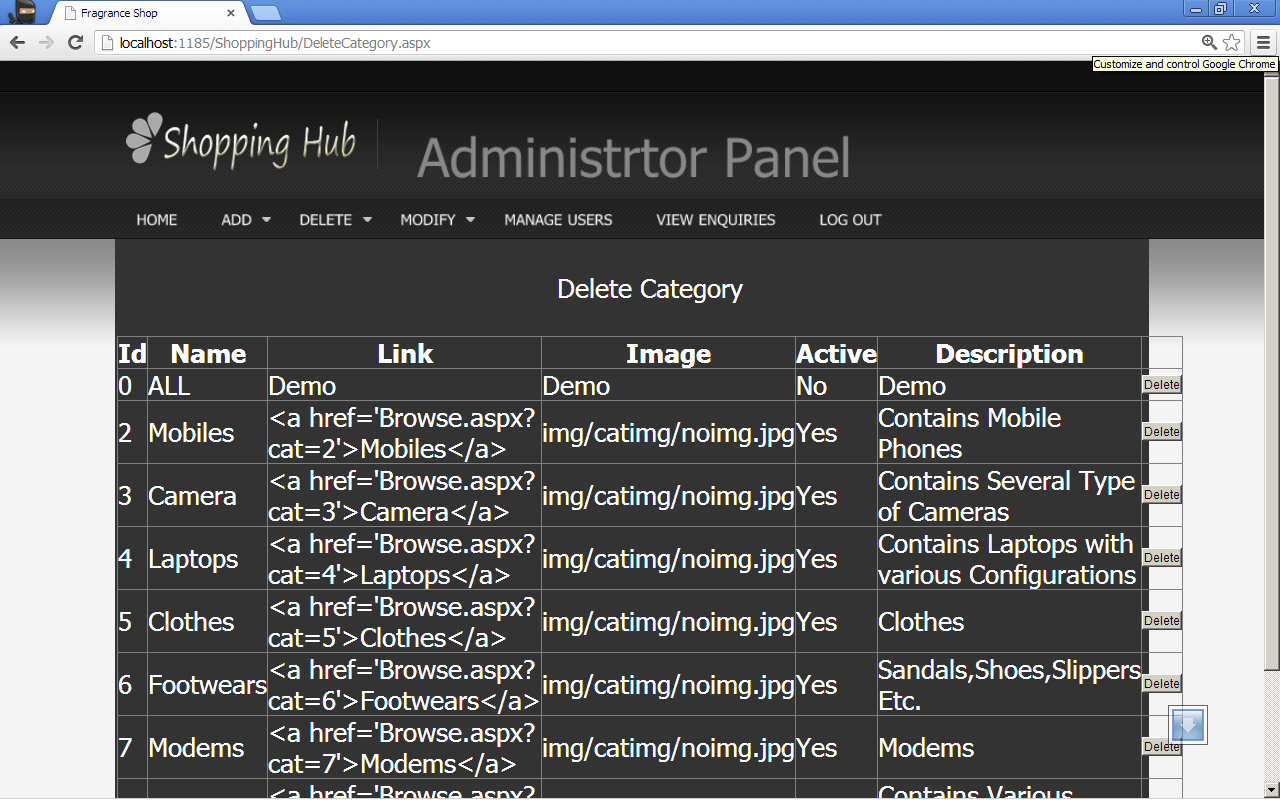
AddNewBrand.aspx



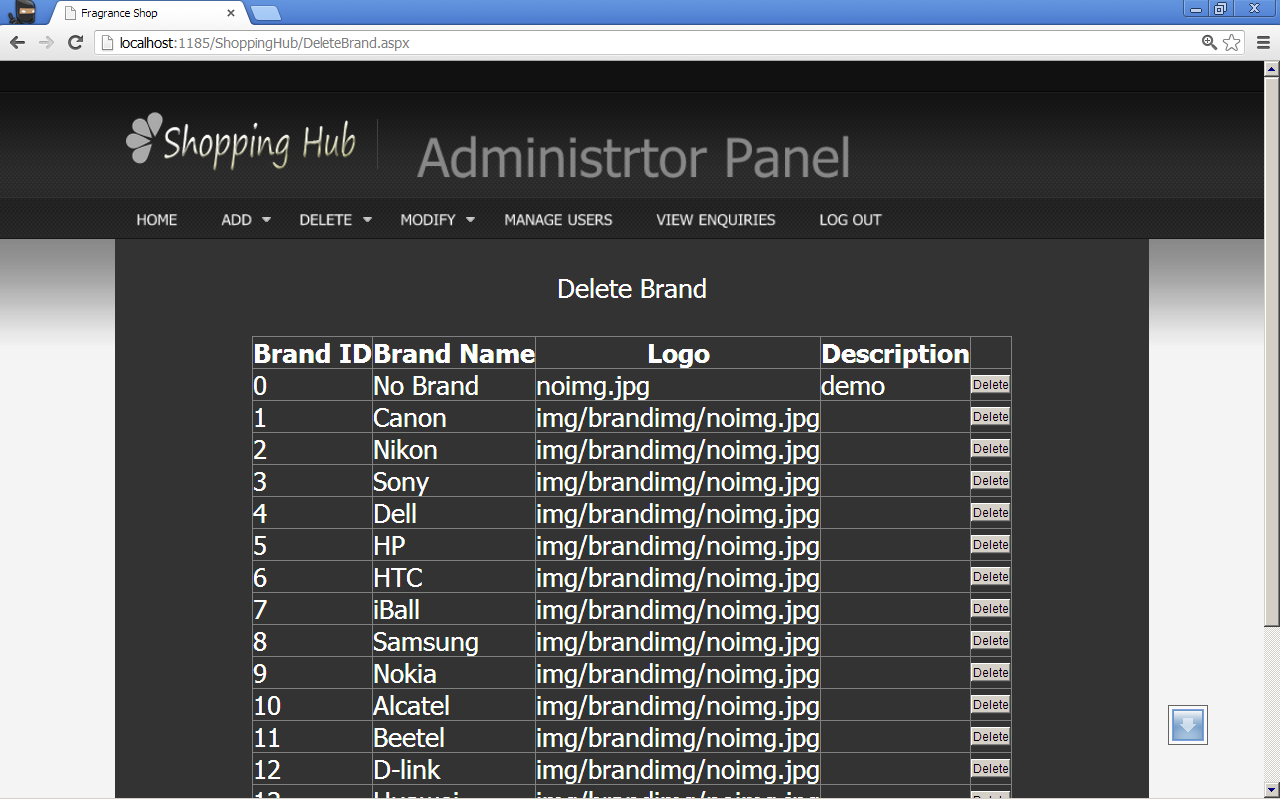
AddNewProduct.aspx



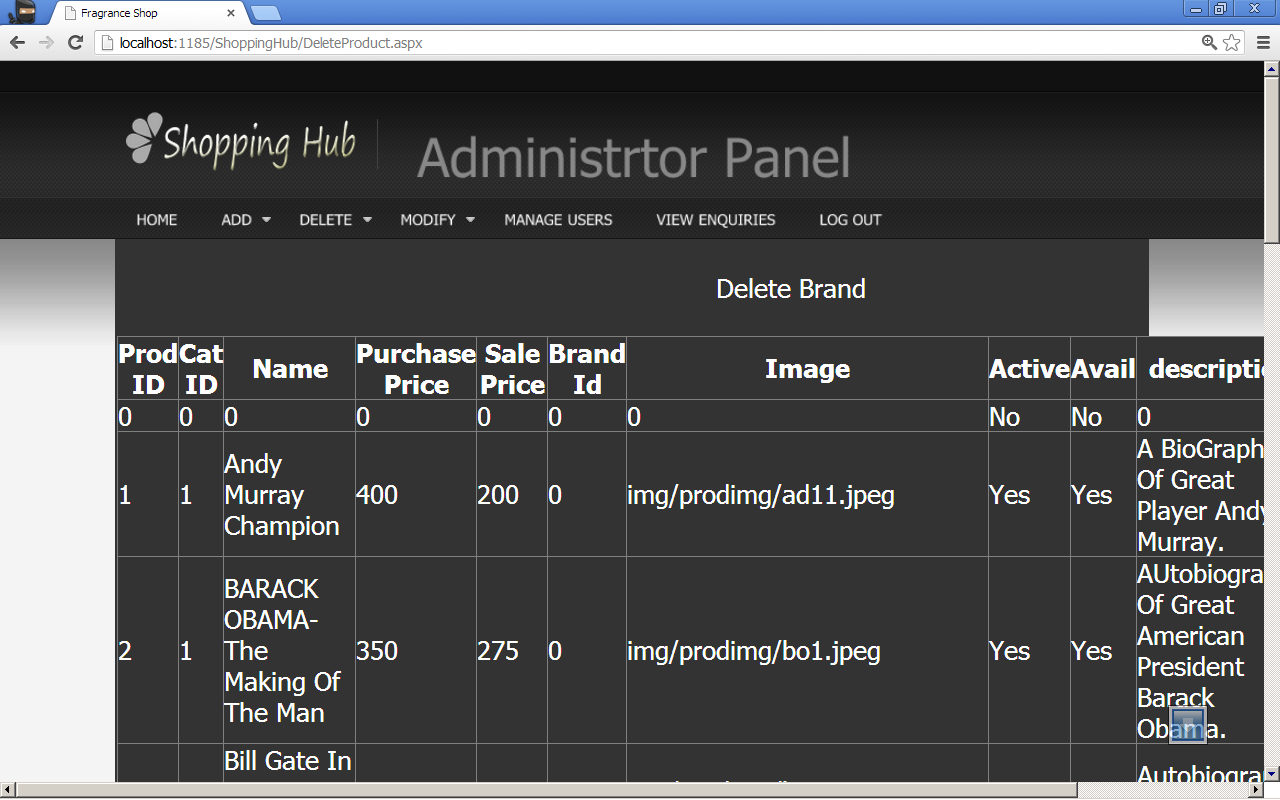
DeleteCategory.aspx



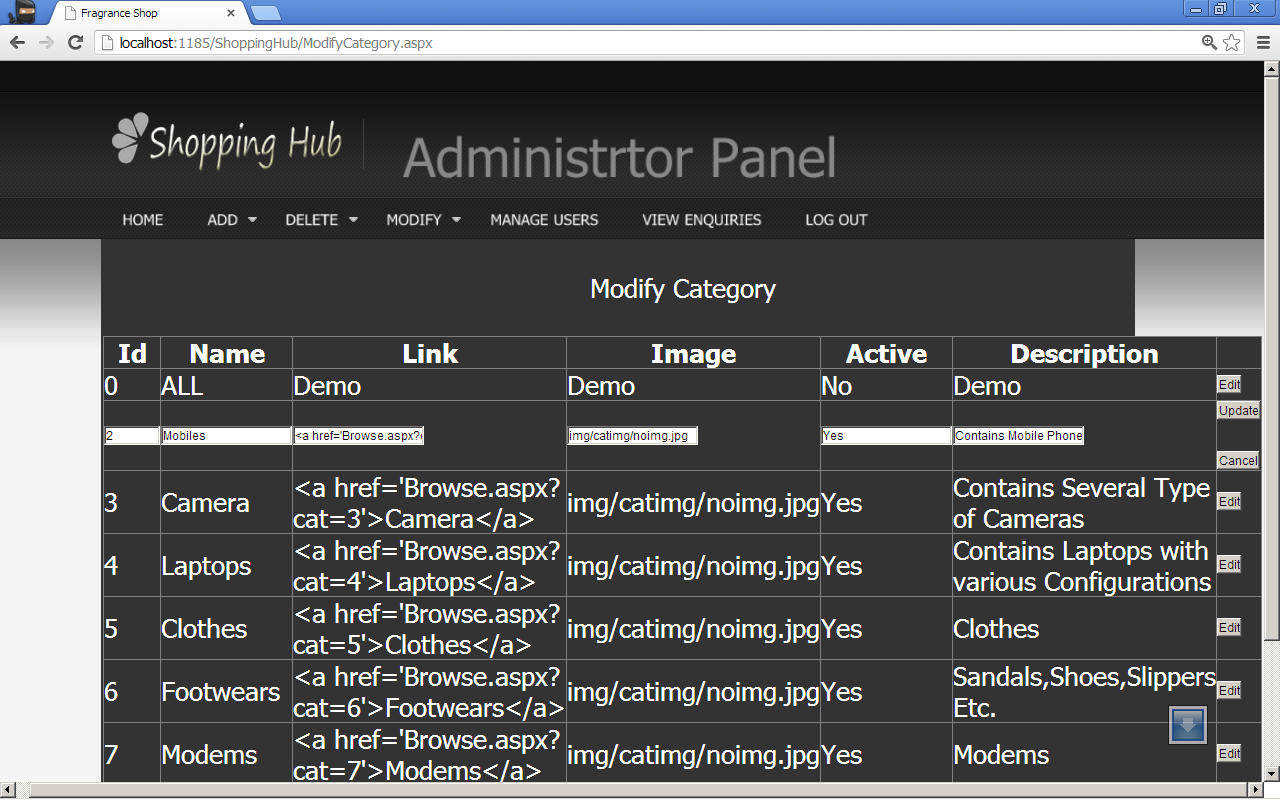
DeleteBrand.aspx



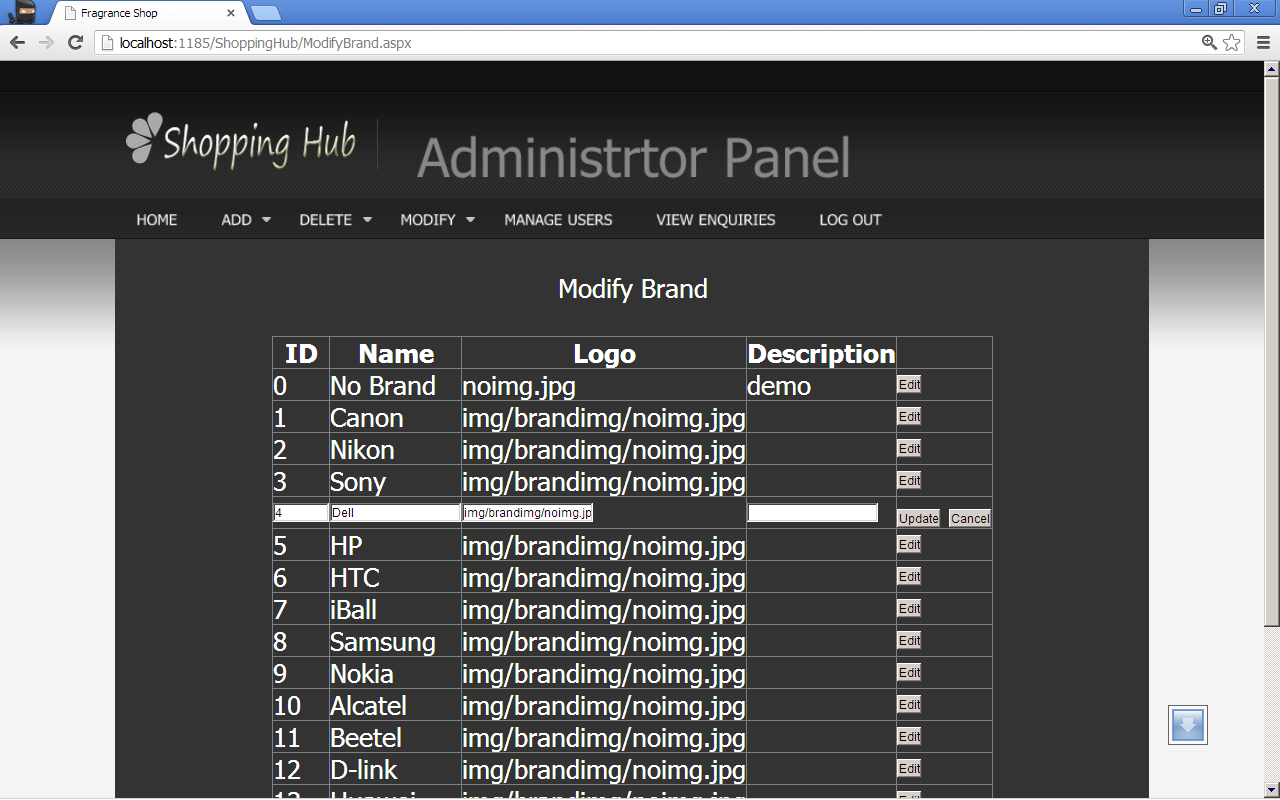
DeleteProduct.aspx



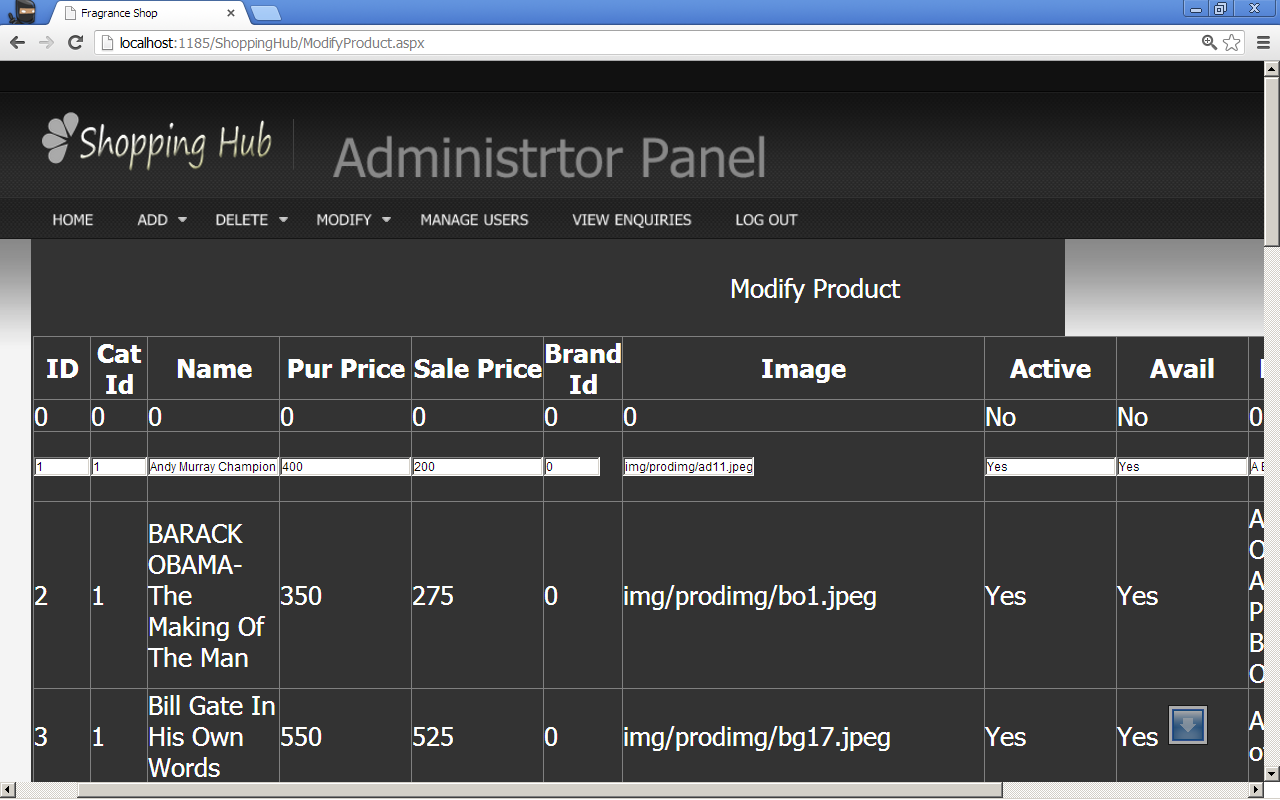
ModifyCategory.aspx



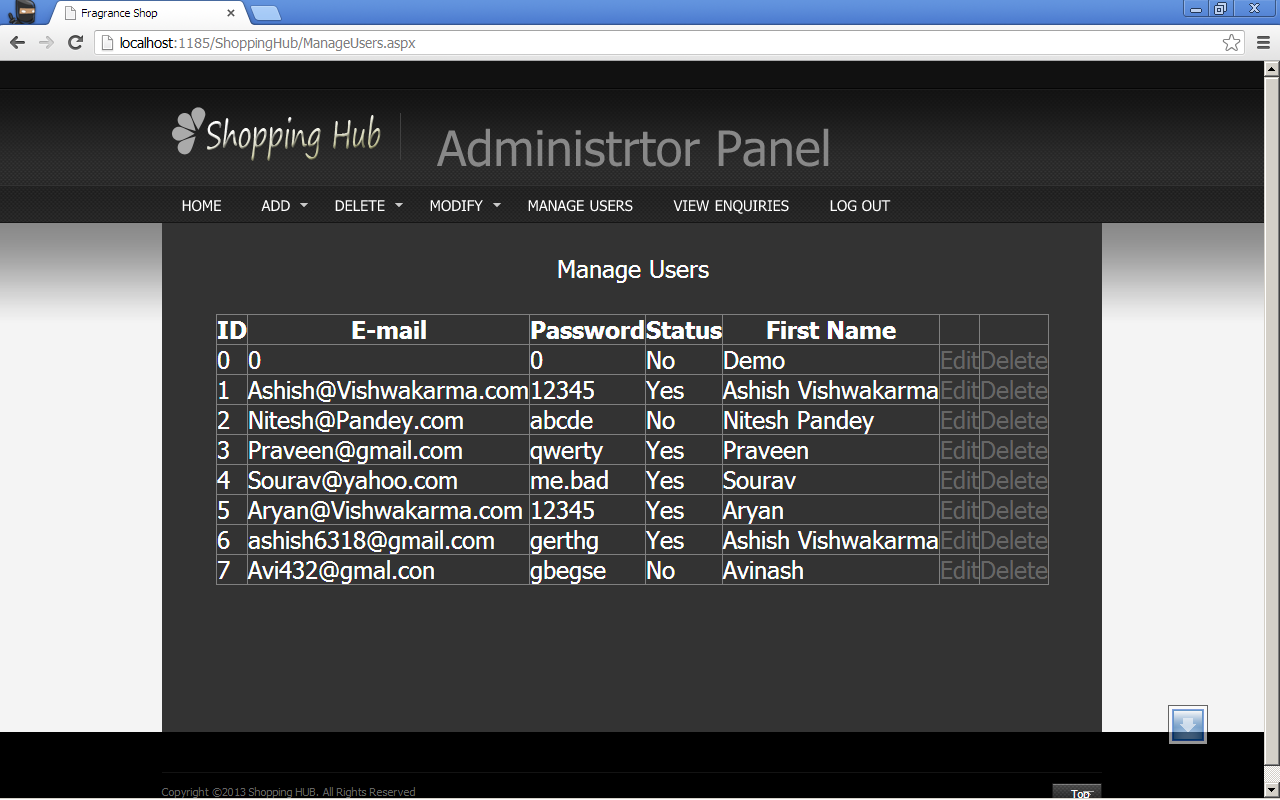
ModifyBrand.aspx



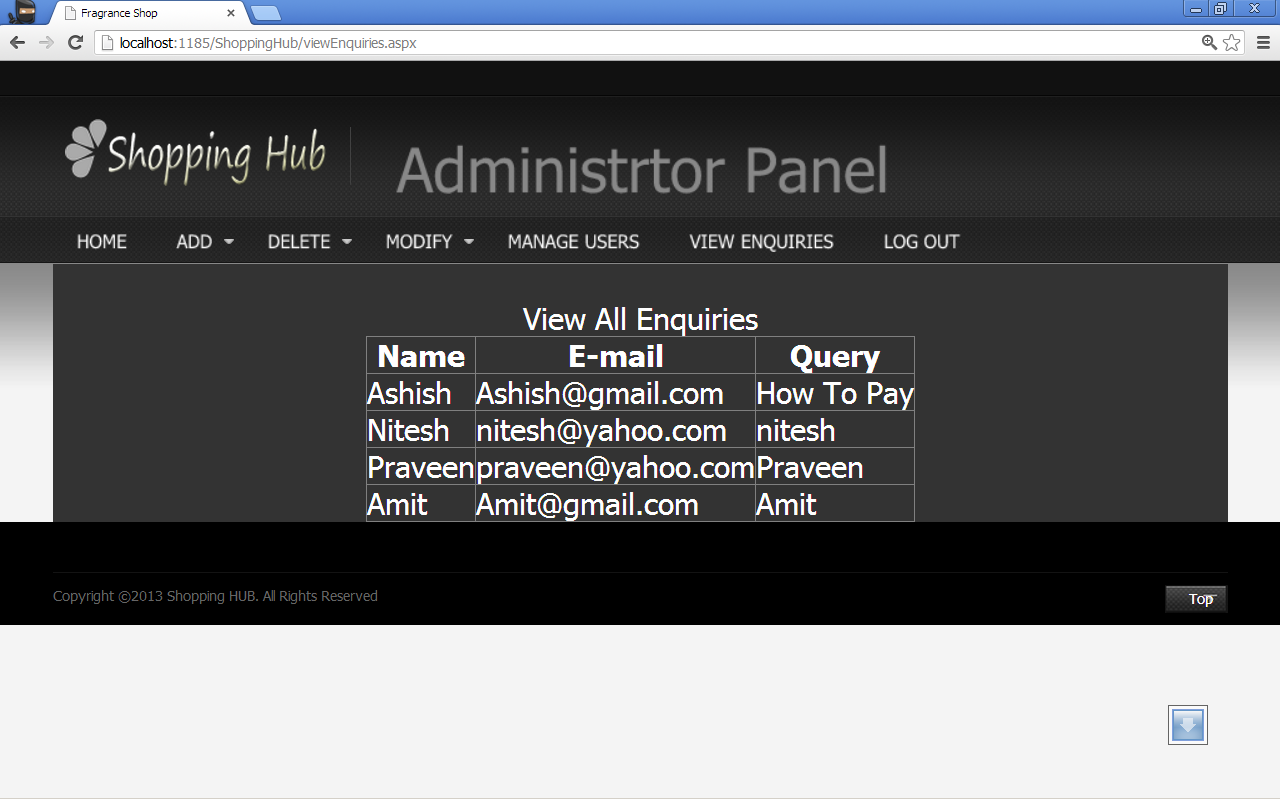
ModifyProduct.aspx



ManageUsers.aspx



ViewEnquiries.aspx



Default.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Web.UI.HtmlControls;

public partial class \_Default : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //making connection object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //establishing connection to database

//Checking if No user is logged in and Showing User menu according to user state

if ((Session["uid"] == "") || (Session["uid"] == null))

{

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

string usr = "select fname from useracc where uid='"+Convert.ToInt32(Session["uid"])+"'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

try

{

// Retrieving all the categories from database and showing them in Datalist Control as Left menu

string str = "select catid,catname,catlink,active from category where active='Yes'";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds, "category");

con.Close();

string[,] dytab = new string[9, 9];

for (int i = 0; i < ds.Tables["category"].Rows.Count; i++)

{

dr = ds.Tables["category"].Rows[i];

string cid = Convert.ToString(dr["catid"]);

string cname = Convert.ToString(dr["catname"]);

string clink = Convert.ToString(dr["catlink"]);

//retrieving featured product list from all the categories which are active

string str1 = "select prodid,prodname,saleprice,brandimg from prodtab where catid='" + cid + "' and active='Yes' and avail='Yes' and featured='Yes'";

SqlDataAdapter da1 = new SqlDataAdapter(str1, con);

DataSet ds1 = new DataSet();

DataRow dr1;

con.Open();

da1.Fill(ds1, "product");

con.Close();

for (int k = 0; k < ds1.Tables["product"].Rows.Count; k++)

{

for (int j = 0; j < ds1.Tables["product"].Columns.Count; j++)

{

dr1 = ds1.Tables["product"].Rows[k];

dytab[k, j] = Convert.ToString(dr1[j]);

Label1.Text += dytab[k, j];

}

}

//Displaying all the retrieved product according to categories in stacked tabular form

int catid = Convert.ToInt32(cid);

Label featuredboxlbl = new Label();

//creating category wise featured product tab dynamically

featuredboxlbl.Text = "<table width='100%' border='3' id='mainprodcontainer'> <tr> <td><table width='100%' > <tr> <td align='left'><div id='prodhead'>" + clink + "</div></td> </tr> <tr> <td><table width='100%' ><tr> <td><table width='100%' id='prodcantainer'><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[0, 0] + "'><img src=" + dytab[0, 3] + " /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[0, 0] + "'>" + dytab[0, 1] + "</a></td></tr><tr><td align='center'><a href='WishList.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtowishlist.gif' alt='Add Product To WishList' /></a></td></tr><tr><td align='center'><a href='Cart.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtocart.gif' alt='Add Product To Cart' /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[0, 0] + "'>Rs." + dytab[0, 2] + "</a></td></tr></table></td><td><table width='100%' id='prodcantainer'><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[1, 0] + "'><img src=" + dytab[1, 3] + " /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[1, 0] + "'>" + dytab[1, 1] + "</a></td></tr><tr><td align='center'><a href='WishList.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtowishlist.gif' alt='Add Product To WishList' /></a></td></tr><tr><td align='center'><a href='Cart.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtocart.gif' alt='Add Product To Cart' /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[1, 0] + "'>Rs." + dytab[1, 2] + "</a></td></tr></table></td><td><table width='100%' id='prodcantainer'><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[2, 0] + "'><img src=" + dytab[2, 3] + " /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[2, 0] + "'>" + dytab[2, 1] + "</a></td></tr><tr><td align='center'><a href='WishList.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtowishlist.gif' alt='Add Product To WishList' /></a></td></tr><tr><td align='center'><a href='Cart.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtocart.gif' alt='Add Product To Cart' /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[2, 0] + "'>Rs." + dytab[2, 2] + "</a></td></tr></table></td><td><table width='100%' id='prodcantainer'><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[3, 0] + "'><img src=" + dytab[3, 3] + " /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[3, 0] + "'>" + dytab[3, 1] + "</a></td></tr><tr><td align='center'><a href='WishList.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtowishlist.gif' alt='Add Product To WishList' /></a></td></tr><tr><td align='center'><a href='Cart.aspx?prodid=" + dytab[0, 0] + "'><img src='img/addtocart.gif' alt='Add Product To Cart' /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dytab[3, 0] + "'>Rs." + dytab[3, 2] + "</a></td></tr></table></td> </tr> </table></td> </tr></table></td> </tr> </table><br/>";

featuredboxlbl.ID = "FeaturedBoxLbl" + i.ToString();

PlaceHolder1.Controls.Add(featuredboxlbl);

}

}

//handing th exceptions

catch (SqlException sqex)

{

Label1.Text = sqex.Message;

}

catch (Exception ex)

{

Label1.Text = ex.Message;

}

}

}

Browse.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.Data.SqlClient;

using System.Configuration;

using System.Web.UI.HtmlControls;

public partial class Browse : System.Web.UI.Page

{

SqlConnection con = new SqlConnection();

//creating connection object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //creating connection to database

string catid = Request.QueryString["cat"];

//retrieving category id from querystring to show product of that category in page

string[] dyt = new string[5];

//Checking if No user is logged in and Showing User menu according to user state

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//dynamically creating user menu if user is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//retrieving user's name according session uid is user is logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

//dynamically creating user menu if user is logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

try

{

//retrieving all the product of category which is passed through querystring

string str = "select prodid,prodname,saleprice,brandimg from prodtab where catid='"+catid+"'";

SqlDataAdapter da = new SqlDataAdapter(str,con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"product");

con.Close();

for (int k = 0; k < ds.Tables["product"].Rows.Count; k++)

{

for (int j = 0; j < ds.Tables["product"].Rows.Count; j++)

{

dr = ds.Tables["product"].Rows[k];

dyt[j] = Convert.ToString(dr[j]);

}

Label singleprod = new Label();

//creating product tab dynamically within div tag which are flotted left

singleprod.Text = "<div style='float:left;'><table width='170px' height='250px' id='singleprodcantainer'><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dyt[0] + "'><img src=" + dyt[3] + " /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dyt[0] + "'>" + dyt[1] + "</a></td></tr><tr><td align='center'><a href='WishList.aspx?prodid=" + dyt[0] + "'><img src='img/addtowishlist.gif' alt='Add Product To WishList' /></a></td></tr><tr><td align='center'><a href='Cart.aspx?prodid=" + dyt[0] + "'><img src='img/addtocart.gif' alt='Add Product To Cart' /></a></td></tr><tr><td align='center'><a href='ProdDetails.aspx?prodid=" + dyt[0] + "'>Rs." + dyt[2] + "</a></td></tr></table></div>";

singleprod.ID = "SingleProduct" + k.ToString();

PlaceHolder1.Controls.Add(singleprod);

}

}

//handling exception

catch (SqlException ex)

{

Label1.Text = ex.Message;

}

catch (Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

ProdDetails.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required name space for database connection

using System.Data.SqlClient; //using required name space for database connection

using System.Configuration; //using required name space for database connection

using System.Web.UI.HtmlControls;

public partial class ProdDetails : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //creating database connection object

protected void Page\_Load(object sender, EventArgs e)

{

try

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //establishing connection to database

string pid = Request.QueryString["prodid"]; //retrieving product id from querystring to show specific product details on page

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//dynamically creating user menu if user is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//retrieving user's name according session uid is user is logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

//dynamically creating user menu if user is logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

//retrieving product's detail from database according to product id passed from query string

string str = "select prodname,saleprice,brandimg,description from prodtab where prodid='" + pid + "'";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds, "prod");

con.Close();

dr = ds.Tables["prod"].Rows[0];

string pname = Convert.ToString(dr[0]);

string price = Convert.ToString(dr[1]);

string img = Convert.ToString(dr[2]);

string des = Convert.ToString(dr[3]);

//arranging product details in dynamically created table retrieved from database

Label pDetails = new Label();

pDetails.Text = "<table width='100%'><tr><td colspan='2'></td></tr><tr><td colspan='2' align='center'><h5 class='small\_head'>" + pname + "</h5></td></tr><tr><td rowspan='2' align='right'><img src=" + img + "/></td><td align='left'><h2>Rs." + price + "</h2></td></tr><tr><td align='left'><h2>" + des + "</h2></td></tr><tr><td align='right'><a href='Wishlist.aspx?prodid=" + pid + "'><img src='img/addtowishlist.gif' /></a></td><td align='left'><a href='Cart.aspx?prodid=" + pid + "'><img src='img/addtocart.gif' /></a></td></tr></table>";

pDetails.ID = "ProductDetails";

PlaceHolder1.Controls.Add(pDetails);

}

//handling exception

catch(SqlException ex)

{

Label1.Text=ex.Message;

}

catch(Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

Cart.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespaces for database connectivity

using System.Data.SqlClient; //using required namespaces for database connectivity

using System.Configuration; //using required namespaces for database connectivity

using System.Web.UI.HtmlControls;

public partial class Cart : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //making connection object

protected void Page\_Load(object sender, EventArgs e)

{

List<string> ls = (List<string>)Application.Get("Cart"); //retrieving Application State variable 'Cart' from Global.asax to List ls

int total = 0; //variable for storing total amount to pay

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //connecting to database via connection string

try

{

//making usermenu according to User State

if (Session["uid"] != "" || Session != null)

{

//Usermenu is yser is logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da5 = new SqlDataAdapter(usr, con);

DataSet ds5 = new DataSet();

DataRow dr5;

con.Open();

da5.Fill(ds5, "name");

con.Close();

dr5 = ds5.Tables["name"].Rows[0];

string nm = Convert.ToString(dr5["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//user menu if user is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

//Cheking id product id is passed with Query String

if ((Request.QueryString["prodid"] == "") || (Request.QueryString["prodid"]) == null)

{

foreach(string p in ls)

{

//if no Poduct rae added to cart Showing message that cart is Empty

if (p == null || p == "")

{

Label empt = new Label();

empt.Text = "Your Cart Is Empty";

PlaceHolder1.Controls.Add(empt);

}

else

{

//if no prduct id is Passed via query string and cart has products, then it Shows Items available in Cart

string str="select prodname,saleprice,brandimg from prodtab where prodid='"+p+"'";

SqlDataAdapter da=new SqlDataAdapter(str,con);

DataSet ds=new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"product");

con.Close();

dr = ds.Tables["product"].Rows[0];

string prodname = Convert.ToString(dr["prodname"]);

string saleprice = Convert.ToString(dr["saleprice"]);

string img = Convert.ToString(dr["brandimg"]);

Label cartItem = new Label();

//Listing Cart's Product in dynamically created table

cartItem.Text = "<table width='100%' align='center' border='1'><tr><td><table align='center'><tr><td rowspan='2'><img src='"+img+"'/></td><td>"+prodname+"</td><td rowspan='2'>Prod</td><td rowspan='2'><a href='RemoveFromCart.aspx?removeProdID="+p+"'>Remove From Cart</a></td></tr><tr><td>"+saleprice+"</td></tr></table></td></tr></table>";

PlaceHolder1.Controls.Add(cartItem);

total += Convert.ToInt32(saleprice);

}

}

//Showing Total Amount to Pay in dynamicallt created Label.

Label Gtotal = new Label();

Gtotal.Text = "<table width='100%'><tr><td><div align='center'>" + total + "</div></td></tr></table>";

PlaceHolder3.Controls.Add(Gtotal);

Application["Total"] = total; //Copying Total AMount to Pay in Application State Variable 'Total'

}

else

{

//if Product id is passed via query string it is added to Shopping Cart

string prodid = Convert.ToString(Request.QueryString["prodid"]);

ls.Add(prodid);

//Retriving Product Details from Database which is added to cart

string str2 = "select prodname,saleprice,brandimg from prodtab where prodid='" +prodid+ "'";

SqlDataAdapter da2 = new SqlDataAdapter(str2, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2,"AddProd");

con.Close();

dr2=ds2.Tables["AddProd"].Rows[0];

string prodname = Convert.ToString(dr2["prodname"]);

string saleprice = Convert.ToString(dr2["saleprice"]);

string img = Convert.ToString(dr2["brandimg"]);

Label cartItem = new Label();

//Showing product detail which is retrieved from database and added to cart

cartItem.Text = "<table width='100%' align='center' border='1'><tr><td><table align='center'><tr><td rowspan='2'><img src='" + img + "'/></td><td>" + prodname + "</td><td rowspan='2'>Prod</td><td rowspan='2'><a href='RemoveFromCart.aspx?removeProdID=" + prodid + "'>Remove From Cart</a></td></tr><tr><td>" + saleprice + "</td></tr></table></td></tr></table>";

PlaceHolder1.Controls.Add(cartItem);

Label1.Text =prodname+" SuccessFully Added To Cart !";

}

}

//handling excedption

catch(SqlException ex)

{

Label1.Text = ex.Message;

}

catch(Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

SignUp.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class user\_AddNewUser : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //Connecting to database via global connectioo string

//Checking User State

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//Showing user menu if User is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder1.Controls.Add(userMenu);

}

else

{

//if user is already logged in he/she dosen't need to sign up so redirect to HomePage

Response.Redirect("Default.aspx");

}

//Executing Stored procedure for auto generating user id so they will be unique

string str = "exec useridautogen";

SqlDataAdapter da = new SqlDataAdapter(str,con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"#top");

con.Close();

dr = ds.Tables["#top"].Rows[0];

TextBox1.Text = dr["var"].ToString(); //putting auto generated uid to textbox1 which is read only

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

//executing stored procedure for inserting credentials to database

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "usrsignup";

cmd.Parameters.AddWithValue("@uid", Convert.ToInt32(TextBox1.Text)); //assigning values to stored procedure for inserting in database

cmd.Parameters.AddWithValue("@email", TextBox3.Text.Trim()); //assigning values to stored procedure for inserting in database

cmd.Parameters.AddWithValue("pwd", TextBox4.Text.Trim()); //assigning values to stored procedure for inserting in database

cmd.Parameters.AddWithValue("@status", "Yes"); //assigning values to stored procedure for inserting in database

cmd.Parameters.AddWithValue("@fname", TextBox2.Text.Trim()); //assigning values to stored procedure for inserting in database

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Session["uid"] = TextBox1.Text.Trim();

Response.Redirect("Default.aspx");

TextBox2.Text = "";

TextBox3.Text = "";

TextBox4.Text = "";

TextBox5.Text = "";

}

//handling exceptions

catch (SqlException ex)

{

Label4.Text = ex.Message;

}

catch (Exception ex1)

{

Label4.Text = ex1.Message;

}

}

}

ChangePassword.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //Using Required namespace

using System.Data.SqlClient; //Using Required namespace

using System.Configuration; //Using Required namespace

public partial class ChangePassword : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //establishing connection to database

//Checking if No user is logged in and Showing User menu according to user state

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//Usermenu if User is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//user menu if usaer in logged in

//retriving user name if user is logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

//Executing Procedure chpwd to change password

string str = "exec chpwd '"+Convert.ToInt32(Session["uid"])+"','"+TextBox1.Text+"','"+TextBox3.Text+"'";

SqlDataAdapter da = new SqlDataAdapter(str,con);

DataSet ds=new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"#temp");

con.Close();

dr = ds.Tables["#temp"].Rows[0];

//Getting message via flag variable a

int a = Convert.ToInt32(dr["var"]);

if (a == 1)

{

//if flag is 1 then password is changed

Label1.Text = "Password Changed!";

}

else

{

//if flag is 0 then password is not changed

Label1.Text = "Password Not Changed!";

}

}

}

UserHome.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace

using System.Data.SqlClient; //using required namespace

using System.Configuration; //using required namespace

public partial class UserHome : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //connecting to database via global connectionstring

//Checking User state if user is logged in

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//if user is not logged redirect to Sign Up Page

Response.Redirect("SignUp.aspx");

}

else

{

//If User Is logged in showing Usermenu accordingly

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

//Showing Welcome message with User Name

Label uname = new Label();

uname.Text = "Welcome " + nm + "!";

uname.ID = "uname";

PlaceHolder1.Controls.Add(uname);

//Retrieving and Showing No. of items in Wishlist from database

string str2 = "select count(uid) from wishlist where uid='"+Convert.ToInt32(Session["uid"])+"'";

SqlDataAdapter da = new SqlDataAdapter(str2,con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds, "wilist");

con.Close();

dr=ds.Tables["wilist"].Rows[0];

Label wilist = new Label();

wilist.Text = "You Have "+Convert.ToString(dr[0])+" Items in Your Wishlist !";

wilist.ID = "wilist";

PlaceHolder3.Controls.Add(wilist);

//Retrieving and Showing No. of items in Shopping Cart From Application State Variable

List<string> ls=(List<string>)Application.Get("Cart");

int c=Convert.ToInt32(ls.Count);

Label cart = new Label();

cart.Text = "You Have "+c+" Items in Your Cart !";

cart.ID = "Cart";

PlaceHolder4.Controls.Add(cart);

}

}

}

WishList.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //Using required namespace

using System.Data.SqlClient; //Using required namespace

using System.Configuration; //Using required namespace

public partial class wishlist : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

try

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //Connecting to database via global connection string

//Checking if User is logged in

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//if User is not logged in redirected to Sign Up Page

Response.Redirect("SignUp.aspx");

}

else

{

//if User is logged in Showing Menu According

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da5 = new SqlDataAdapter(usr, con);

DataSet ds5 = new DataSet();

DataRow dr5;

con.Open();

da5.Fill(ds5, "name");

con.Close();

dr5 = ds5.Tables["name"].Rows[0];

string nm = Convert.ToString(dr5["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

//Checking if Product ID is passed via query string or not

if (Request.QueryString["prodid"] == "" || Request.QueryString["prodid"] == null)

{

//if product id is not passed via query string then it shows already existed Products

int prodid = Convert.ToInt32(Request.QueryString["prodid"]);

//retrieving all product id from wishlist table according to uid passed in session

string str2 = "select prodid,datead from wishlist where uid='"+Convert.ToInt32(Session["uid"])+"'";

SqlDataAdapter da2 = new SqlDataAdapter(str2, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "prodlist");

con.Close();

for (int i = 0; i < ds2.Tables["prodlist"].Rows.Count; i++)

{

//retrieving Product details from product table accroding to uid retrieved from last query

dr2 = ds2.Tables["prodlist"].Rows[i];

int pid=Convert.ToInt32(dr2["prodid"]);

string dt = Convert.ToString(dr2["datead"]);

string str3 = "select prodname,saleprice,brandimg from prodtab where prodid='"+pid+"'";

SqlDataAdapter da3= new SqlDataAdapter(str3,con);

DataSet ds3= new DataSet();

DataRow dr3;

con.Open();

da3.Fill(ds3,"wlist");

con.Close();

for (int j = 0; j < ds3.Tables["wlist"].Rows.Count; j++)

{

dr3 = ds3.Tables["wlist"].Rows[j];

string pnm = Convert.ToString(dr3["prodname"]);

string price = Convert.ToString(dr3["saleprice"]);

string img = Convert.ToString(dr3["brandimg"]);

//Dynamically Showing already exised Content of wishlist

Label prod = new Label();

prod.Text = "<table align='center' border='1' width='100%'><tr><td><img src='"+img+"'/></td><td>"+pnm+"</td><td>"+price+"</td></tr></table><br/>";

PlaceHolder1.Controls.Add(prod);

}

}

}

else

{

//iF Product ID is passed via query string then it is added to WIshlist table with uid passed via Session

int prodid = Convert.ToInt32(Request.QueryString["prodid"]);

//Procedure for auto Generating Wishlist ID

string str = "exec wlidautogen";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds, "#top");

con.Close();

dr = ds.Tables["#top"].Rows[0];

int wlid = Convert.ToInt32(dr["var"]);

int uid = Convert.ToInt32(Session["uid"]);

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

//Procedure for inserting values to wishlist table

cmd.CommandText = "addtowishlist";

cmd.Parameters.AddWithValue("@wlid", wlid);

cmd.Parameters.AddWithValue("@uid", uid);

cmd.Parameters.AddWithValue("@prodid", prodid);

cmd.Parameters.AddWithValue("@datead", Convert.ToString(DateTime.Now));

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

con.Close();

//query for showing details of product which is added to wishlist

string str2 = "select prodname,saleprice,brandimg from prodtab where prodid='" + prodid + "'";

SqlDataAdapter da2 = new SqlDataAdapter(str2, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "AddProd");

con.Close();

dr2 = ds2.Tables["AddProd"].Rows[0];

string prodname = Convert.ToString(dr2["prodname"]);

string saleprice = Convert.ToString(dr2["saleprice"]);

string img = Convert.ToString(dr2["brandimg"]);

//Showing Currently added content in dynamically generated table

Label cartItem = new Label();

cartItem.Text = "<table width='100%' align='center' border='1'><tr><td><table align='center'><tr><td rowspan='2'><img src='" + img + "'/></td><td>" + prodname + "</td><td rowspan='2'>Prod</td><td rowspan='2'><a href='RemoveFromCart.aspx?removeProdID=" + prodid + "'>Remove From Cart</a></td></tr><tr><td>" + saleprice + "</td></tr></table></td></tr></table>";

PlaceHolder1.Controls.Add(cartItem);

Label1.Text = prodname + " SuccessFully Added To Cart !";

} } }

//Handling Exceptions

catch (SqlException ex)

{

Label1.Text = ex.Message;

}

catch (Exception ex1)

{

Label1.Text = ex1.Message;

} }

}

ContactUs.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class ContactUs : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Conection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //connecting to database via global connection string

//Checking if User is Logged In

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//creating userMenu if user is not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//creating UserMenu if user in Logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

//Executing Procedure 'insqry' for inserting values to database

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "insqry";

cmd.Parameters.AddWithValue("@nm",TextBox1.Text.Trim());

cmd.Parameters.AddWithValue("@email", TextBox2.Text.Trim());

cmd.Parameters.AddWithValue("@qry",TextBox3.Text.Trim());

cmd.Connection = con;

try

{

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("ContactUs.aspx");

TextBox1.Text = "";

TextBox2.Text = "";

TextBox3.Text = "";

}

//Handling exception

catch (SqlException ex)

{

Label1.Text = ex.Message;

}

catch (Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

AboutUs.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class AboutUs : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString=ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //establishing connection via Global Connection String

//Checking if User is logged in

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//creating usermenu if user in not Logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

else

{

//creating Usermenu if user is logged in

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

}

}

Payment.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting to database

using System.Data.SqlClient; //using required namespace for connecting to database

using System.Configuration; //using required namespace for connecting to database

public partial class Payment : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //establishing connection to database

//checking is user is logged in

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//if user is not logged in redirected to Sign Up Page

Response.Redirect("Signup.aspx");

}

else

{

//if user is logged in creating menu accordingly

string usr = "select fname from useracc where uid='" + Convert.ToInt32(Session["uid"]) + "'";

SqlDataAdapter da2 = new SqlDataAdapter(usr, con);

DataSet ds2 = new DataSet();

DataRow dr2;

con.Open();

da2.Fill(ds2, "name");

con.Close();

dr2 = ds2.Tables["name"].Rows[0];

string nm = Convert.ToString(dr2["fname"]);

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome " + nm + "!</li><li><a href='UserHome.aspx'>My Account</a></li><li><a href='Wishlist.aspx'>My Wishlist</a></li><li><a href='Cart.aspx'>My Cart</a></li><li><a href='Payment.aspx'>Checkout</a></li><li><a href='LogOut.aspx'>Log Out</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder2.Controls.Add(userMenu);

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

//Executing Procedure 'makeorder' for inserting values to Order Table

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "makeorder";

cmd.Parameters.AddWithValue("@uid", Convert.ToInt32(Session["uid"]));

cmd.Parameters.AddWithValue("@address", TextBox1.Text.Trim());

cmd.Parameters.AddWithValue("@payoption", RadioButtonList1.Text);

cmd.Connection = con;

con.Open();

cmd.ExecuteNonQuery();

con.Close();

//forwarding detail to paypal for making Payment

Response.Redirect("http://paypal.com?id=XXXXpayoption=XXX");

}

//Handling Exceptions

catch(SqlException ex)

{

Label4.Text=ex.Message;

}

catch(Exception ex1)

{

Label4.Text = ex1.Message;

}

}

}

UserLogin.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for Connecting to database

using System.Data.SqlClient; //using required namespace for Connecting to database

using System.Configuration; //using required namespace for Connecting to database

public partial class SignIn : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //connecting to database via global connection string

//Checking if user is logged in

if ((Session["uid"] == "") || (Session["uid"] == null))

{

//UserMenu if user not logged in

Label userMenu = new Label();

userMenu.Text = "<ul class='links'><li class='bold first'>Welcome Guest!</li><li><a href='SignUp.aspx'>My Account</a></li><li><a href='SignUp.aspx'>My Wishlist</a></li><li><a href='SignUp.aspx'>My Cart</a></li><li><a href='SignUp.aspx'>Checkout</a></li><li><a href='UserLogin.aspx'>Log In</a></li></ul>";

userMenu.ID = "userMenu";

PlaceHolder1.Controls.Add(userMenu);

}

else

{

//UserMenu if user in logged in ,no need to sign up,Redirected to HomePage

Response.Redirect("Default.aspx");

}

}

protected void Login1\_Authenticate(object sender, AuthenticateEventArgs e)

{

//executing procedure 'loginpro' for Logging in

string str = "exec loginpro '"+Login1.UserName+"','"+Login1.Password+"'";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

try

{

con.Open();

da.Fill(ds,"#temp");

con.Close();

dr = ds.Tables["#temp"].Rows[0];

int a = Convert.ToInt32(dr["var"]);

if (a == 1)

{

string str1 = "select uid from useracc where email='" + Login1.UserName.ToString() + "'";

SqlDataAdapter da1 = new SqlDataAdapter(str1, con);

DataSet ds1 = new DataSet();

DataRow dr1;

con.Open();

da1.Fill(ds1, "uid");

con.Close();

dr1 = ds1.Tables["uid"].Rows[0];

//placing uid to session variable if login is successfull

Session["uid"] = Convert.ToInt32(dr1["uid"]);

Response.Redirect("Default.aspx");

}

else

{

//Showing message if Login fails

Login1.FailureText = "Invalid Password !";

}

}

//Handling Exceptions

catch(SqlException ex)

{

Label1.Text=ex.Message;

}

catch(Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

RemoveFromCart.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class RemoveFromCart : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//importing Application State variable ‘Cart’ to List ls

List<string> ls = (List<string>)Application.Get("Cart");

string removeProd = Request.QueryString["removeProdID"];

//Removing Product Id From ‘Cart’ Passed by Query String

ls.Remove(removeProd);

//Redirecting to Cart page

Response.Redirect("Cart.aspx");

}

}

LogOut.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class LogOut : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Session["uid"] = ""; //Assigning empty to session uid

Session["uid"] =null;//Assigning null to session uid

Session.Abandon(); //Destrying Session

Response.Redirect("Default.aspx"); //Redirecting to Homepage

}

}

AddNewBrand.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class admin\_AddNewBrand : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting to AdminLogin Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

//Executing Procedure 'brandidautogen' for AutoGenerating Brand ID

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString;

string str = "exec brandidautogen";

SqlDataAdapter da = new SqlDataAdapter(str,con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds, "#top");

con.Close();

dr = ds.Tables["#top"].Rows[0];

TextBox1.Text = dr["var"].ToString();

}

protected void Button1\_Click(object sender, EventArgs e)

{

//Executing Procedure 'insprod' for inserting into Database

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "insbrand";

string logo = "img/brandimg/noimg.jpg";

//Checking if file upload control has file or not

if (FileUpload1.HasFile)

{

string d = (System.IO.Path.GetExtension(FileUpload1.FileName)).ToLower();

//validating filetype

if (d == ".jpg" || d == ".jpeg" || d == ".png")

{

string path = "~/img/brandimg/";

string path2 = Server.MapPath(path);

string f = Convert.ToString(DateTime.Now.GetHashCode()) + d;

FileUpload1.SaveAs(path2 + f);

logo = "img/brandimg/" + f;

}

}

cmd.Parameters.AddWithValue("@brandid",Convert.ToInt32(TextBox1.Text));

cmd.Parameters.AddWithValue("@brandname",TextBox2.Text.Trim());

cmd.Parameters.AddWithValue("@logo",logo);

cmd.Parameters.AddWithValue("@description",TextBox3.Text.Trim());

cmd.Connection = con;

try

{

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("AddNewBrand.aspx");

TextBox1.Text = "";

TextBox2.Text = "";

TextBox3.Text = "";

}

//Handling Exceptions

catch (SqlException ex)

{

Label5.Text = ex.Message;

}

catch (Exception ex1)

{

Label5.Text = ex1.Message;

}

}

}

AddNewCategory.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Web.UI.HtmlControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class admin\_Default : System.Web.UI.Page

{

SqlConnection con = new SqlConnection();

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting to AdminLogin Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

//Executing Procedure 'catidautogen' for AutoGenerating Category ID

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString;

string str = "exec catidautogen";

SqlDataAdapter da = new SqlDataAdapter(str,con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"#top");

con.Close();

dr=ds.Tables["#top"].Rows[0];

TextBox1.Text=dr["var"].ToString();

}

protected void Button1\_Click(object sender, EventArgs e)

{

//Executing Procedure 'inscat' for inserting into Database

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "inscat";

Label6.Text = "<a href='Browse.aspx?cat="+TextBox1.Text.Trim()+"'>"+TextBox2.Text.Trim()+"</a>";

//Checking if file upload control has file or not

if (FileUpload1.HasFile)

{

string d = (System.IO.Path.GetExtension(FileUpload1.FileName)).ToLower();

//validating filetype

if (d == ".jpg" || d == ".jpeg" || d == ".png")

{

string path = "~/img/catimg/";

string path2 = Server.MapPath(path);

string f=Convert.ToString(DateTime.Now.GetHashCode()) + d;

FileUpload1.SaveAs(path2 +f);

Label7.Text = "img/catimg/" + f;

}

}

cmd.Parameters.AddWithValue("@catid", Convert.ToInt32(TextBox1.Text.Trim()));

cmd.Parameters.AddWithValue("@catname", TextBox2.Text.Trim());

string ctnm = TextBox2.Text.Trim();

cmd.Parameters.AddWithValue("@catlink",Label6.Text);

cmd.Parameters.AddWithValue("@catimg",Label7.Text);

cmd.Parameters.AddWithValue("@active", RadioButtonList1.Text);

cmd.Parameters.AddWithValue("@description", TextBox3.Text);

cmd.Connection = con;

try

{

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("AddNewCategory.aspx");

TextBox2.Text = "";

TextBox3.Text = "";

Label8.Text = ctnm + " Successfully Added!";

}

//Handling Exceptions

catch (SqlException ex)

{

Label8.Text = "Error:" + ex.Message;

}

catch (Exception ex1)

{

Label8.Text = ex1.Message;

}

}

}

AddNewProduct.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

using System.Web.UI.WebControls;

public partial class admin\_AddNewProduct : System.Web.UI.Page

{

SqlConnection con = new SqlConnection();

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting to AdminLogin Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

//Executing Procedure 'catidautogen' for AutoGenerating Category ID

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString;

string str = "exec prodidautogen";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

con.Open();

da.Fill(ds,"#top");

con.Close();

dr = ds.Tables["#top"].Rows[0];

TextBox1.Text = dr["var"].ToString();

}

protected void Button1\_Click(object sender, EventArgs e)

{

//Executing Procedure 'insprod' for inserting into Database

SqlCommand cmd = new SqlCommand();

cmd.CommandType = CommandType.StoredProcedure;

cmd.CommandText = "insprod";

Label13.Text = "No";

//Checking if file upload control has file or not

if (FileUpload1.HasFile)

{

string d = (System.IO.Path.GetExtension(FileUpload1.FileName)).ToLower();

//validating filetype

if (d == ".jpg" || d == ".jpeg" || d == ".png")

{

string path = "~/img/prodimg/";

string path2 = Server.MapPath(path);

string f = Convert.ToString(DateTime.Now.GetHashCode()) + d;

FileUpload1.SaveAs(path2 + f);

Label12.Text = "img/prodimg/" + f;

}

}

cmd.Parameters.AddWithValue("@prodid",Convert.ToInt32(TextBox1.Text));

cmd.Parameters.AddWithValue("@catid",Convert.ToInt32(DropDownList1.SelectedValue));

cmd.Parameters.AddWithValue("@prodname",TextBox2.Text.Trim());

cmd.Parameters.AddWithValue("@purprice",TextBox3.Text.Trim()); cmd.Parameters.AddWithValue("@saleprice",TextBox4.Text.Trim());

cmd.Parameters.AddWithValue("@brandid",Convert.ToInt32(DropDownList2.SelectedValue));

cmd.Parameters.AddWithValue("@brandimg", Label12.Text);

cmd.Parameters.AddWithValue("@active",RadioButtonList1.Text);

cmd.Parameters.AddWithValue("@avail",RadioButtonList2.Text);

cmd.Parameters.AddWithValue("@description",TextBox5.Text.Trim());

cmd.Parameters.AddWithValue("@featured", Label13.Text);

cmd.Connection = con;

try

{

con.Open();

cmd.ExecuteNonQuery();

con.Close();

Response.Redirect("AddNewProduct.aspx");

TextBox1.Text = "";

TextBox2.Text = "";

TextBox3.Text = "";

TextBox4.Text = "";

TextBox5.Text = "";

}

//Handling Exceptions

catch (SqlException ex)

{

Label11.Text = ex.Message;

}

catch (Exception ex1)

{

Label11.Text = ex1.Message;

}

}

}

DeleteProduct.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

DeleteCategory.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

DeleteBrand.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

ModifyProduct.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

ModifyBrand.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

ModifyCategory.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

AdminLogOut.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class AdminLogOut : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

Session["Admin"] = "";

Session["Admin"] = null;

Session.Abandon(); //Destroying Session

Response.Redirect("AdminLogin.aspx"); //Redirecting to Admin Login Page

}

}

viewEnquiries.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

AdminHome.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class DeleteProduct : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

//Redirecting To Admin Login Page if Admin is not Logged In

if (Session["Admin"] == null || Session["Admin"] == "")

{

Response.Redirect("AdminLogin.aspx");

}

}

}

AdminLogin.aspx.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data; //using required namespace for connecting database

using System.Data.SqlClient; //using required namespace for connecting database

using System.Configuration; //using required namespace for connecting database

public partial class admLogin : System.Web.UI.Page

{

SqlConnection con = new SqlConnection(); //Making Connection Object

protected void Page\_Load(object sender, EventArgs e)

{

con.ConnectionString = ConfigurationManager.ConnectionStrings["ShoppingHubCS"].ConnectionString; //Connecting toDatabase via global connectionstring

}

protected void Login1\_Authenticate(object sender, AuthenticateEventArgs e)

{

//Executing Stored Procedure 'admloginpro' for Administrator Logging In

string str="exec admloginpro '"+Login1.UserName+"','"+Login1.Password+"'";

SqlDataAdapter da = new SqlDataAdapter(str, con);

DataSet ds = new DataSet();

DataRow dr;

try

{

con.Open();

da.Fill(ds, "#temp");

con.Close();

dr = ds.Tables["#temp"].Rows[0];

int a = Convert.ToInt32(dr["var"]);

if (a == 1)

{

Session["Admin"] = Login1.UserName.ToString();

Response.Redirect("AdminHome");

}

else

{

Login1.FailureText = "Invalid Password";

}

}

//Handling Exception

catch (SqlException ex)

{

Label1.Text = ex.Message;

}

catch (Exception ex1)

{

Label1.Text = ex1.Message;

}

}

}

Global.asax

<%@ Application Language="C#" %>

<script runat="server">

void Application\_Start(object sender, EventArgs e)

{

// Code that runs on application startup

}

void Application\_End(object sender, EventArgs e)

{

// Code that runs on application shutdown

}

void Application\_Error(object sender, EventArgs e)

{

// Code that runs when an unhandled error occurs

}

void Session\_Start(object sender, EventArgs e)

{

// Code that runs when a new session is started

Application.Add("Cart", new List<string>()); //List of String Type Application State Variable for Managing Shopping Cart

Application["Total"] = 0; //Application State variable Storing Total Amount to Pay

}

void Session\_End(object sender, EventArgs e)

{

// Code that runs when a session ends.

// Note: The Session\_End event is raised only when the sessionstate mode

// is set to InProc in the Web.config file. If session mode is set to StateServer

// or SQLServer, the event is not raised.

}

</script>

Web.config

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<connectionStrings>

<add name="ApplicationServices" connectionString="data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|\aspnetdb.mdf;User Instance=true" providerName="System.Data.SqlClient"/>

<add name="ShoppingHubCS" connectionString="Data Source=ASHISH-PC;Initial Catalog=ShoppingHub;User ID=sa;Password=123" providerName="System.Data.SqlClient"/>

</connectionStrings>

<system.web>

<compilation debug="true" targetFramework="4.0"/>

<authentication mode="Forms">

<forms loginUrl="~/Account/Login.aspx" timeout="2880"/>

</authentication>

<membership>

<providers>

<clear/>

<add name="AspNetSqlMembershipProvider" type="System.Web.Security.SqlMembershipProvider" connectionStringName="ApplicationServices" enablePasswordRetrieval="false" enablePasswordReset="true" requiresQuestionAndAnswer="false" requiresUniqueEmail="false" maxInvalidPasswordAttempts="5" minRequiredPasswordLength="6" minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10" applicationName="/"/>

</providers>

</membership>

<profile>

<providers>

<clear/>

<add name="AspNetSqlProfileProvider" type="System.Web.Profile.SqlProfileProvider" connectionStringName="ApplicationServices" applicationName="/"/>

</providers>

</profile>

<roleManager enabled="false">

<providers>

<clear/>

<add name="AspNetSqlRoleProvider" type="System.Web.Security.SqlRoleProvider" connectionStringName="ApplicationServices" applicationName="/"/>

<add name="AspNetWindowsTokenRoleProvider" type="System.Web.Security.WindowsTokenRoleProvider" applicationName="/"/>

</providers>

</roleManager>

</system.web>

<system.webServer>

<modules runAllManagedModulesForAllRequests="true"/>

</system.webServer>

</configuration>

Stored Procedure ‘addtowishlist’

ALTER PROCEDURE addtowishlist(@wlid int,@uid int,@prodid int,@datead nvarchar(50))

AS

insert into wishlist(wlid,uid,prodid,datead) values(@wlid,@uid,@prodid,@datead)

RETURN

Stored Procedure ‘admloginpro’

ALTER PROCEDURE admloginpro(@uid nvarchar(50),@pwd nvarchar(50))

AS

declare @pwd1 nvarchar(50)

declare @a int

select @pwd1=pwd from useracc where uid=@uid

if @pwd1=@pwd

begin

set @a=1

end

else

begin

set @a=0

end

create table #temp(var int)

insert into #temp(var) values(@a)

select var from #temp

drop table #temp

RETURN

Stored Procedure ‘brandidautogen’

ALTER PROCEDURE brandidautogen

AS

declare @a int

select @a=max(brandid) from brand

set @a=@a+1

create table #top(var int)

insert into #top(var) values(@a)

select var from #top

drop table #top

RETURN

Stored Procedure ‘catidautogen’

ALTER PROCEDURE catidautogen

AS

declare @a int

select @a=max(catid) from category

set @a=@a+1

create table #top(var int)

insert into #top(var) values(@a)

select var from #top

drop table #top

RETURN

Stored Procedure ‘chpwd’

ALTER PROCEDURE chpwd(@uid int,@oldpwd nvarchar(50),@newpwd nvarchar(50))

AS

declare @pwd1 nvarchar(50)

declare @f int

select @pwd1=pwd from useracc where uid=@uid

if(@pwd1=@oldpwd)

begin

update useracc set pwd=@newpwd where uid=@uid

set @f=1

end

else

begin

set @f=0

end

create table #temp(var int)

insert into #temp(var) values(@f)

select var from #temp

drop table #temp

RETURN

Stored Procedure ‘insbrand’

ALTER PROCEDURE insbrand(@brandid int,@brandname nvarchar(50),@logo nvarchar(50),@description nvarchar(100))

AS

insert into brand(brandid,brandname,logo,description) values(@brandid,@brandname,@logo,@description)

RETURN

Stored Procedure ‘inscat’

ALTER PROCEDURE inscat(@catid int, @catname nvarchar(50), @catlink nvarchar(50), @catimg nvarchar(50), @active nvarchar(3), @description nvarchar(100))

as

insert into category(catid,catname,catlink,catimg,active,description) values(@catid,@catname,@catlink,@catimg,@active,@description)

return

Stored Procedure ‘insprod’

ALTER procedure insprod(@prodid int,@catid int,@prodname nvarchar(50),@purprice nvarchar(50),@saleprice nvarchar(50),@brandid int,@brandimg nvarchar(50),@active nvarchar(3),@avail nvarchar(3),@description nvarchar(100),@featured nvarchar(3))

as

insert into prodtab values(@prodid,@catid,@prodname,@purprice,@saleprice,@brandid,@brandimg,@active,@avail,@description,@featured)

Stored Procedure ‘insqry’

ALTER PROCEDURE insqry(@nm nvarchar(50),@email nvarchar(50),@qry nvarchar(100))

AS

insert into contactqry(nm,email,qry) values(@nm,@email,@qry)

RETURN

Stored Procedure ‘loginpro’

ALTER PROCEDURE loginpro(@uid int,@pwd nvarchar(50))

AS

declare @pwd1 nvarchar(50)

declare @a int

select @pwd1=pwd from useracc where uid=@uid

if @pwd1=@pwd

begin

set @a=1

end

else

begin

set @a=0

end

create table #temp(var int)

insert into #temp(var) values(@a)

select var from #temp

drop table #temp

RETURN

Stored Procedure ‘makeorder’

ALTER PROCEDURE makeorder(@uid int,@address nvarchar(100),@payoption nvarchar(50))

AS

insert into tblorder(uid,address,payoption) values(@uid,@address,@payoption)

RETURN

Stored Procedure ‘prodidautogen’

ALTER procedure prodidautogen

as

declare @a int

select @a=max(prodid) from prodtab

set @a=@a+1

create table #top(var int)

insert into #top(var) values(@a)

select var from #top

drop table #top

RETURN

Stored Procedure ‘useridautogen’

ALTER PROCEDURE useridautogen

AS

declare @a int

select @a=max(uid) from useracc

set @a=@a+1

create table #top(var int)

insert into #top(var) values(@a)

select var from #top

drop table #top

RETURN

Stored Procedure ‘usrsignup’

ALTER PROCEDURE usrsignup(@uid int,@email nvarchar(50),@pwd nvarchar(50),@status nvarchar(3),@fname nvarchar(50))

AS

insert into useracc(uid,email,pwd,status,fname) values(@uid,@email,@pwd,@status,@fname)

RETURN

Stored Procedure ‘wlidautogen’

ALTER PROCEDURE wlidautogen

AS

declare @a int

select @a=max(wlid) from wishlist

set @a=@a+1

create table #top(var int)

insert into #top(var) values(@a)

select var from #top

drop table #top

RETURN

Testing

Software testing is a critical element of software quality assurance and represent the ultimate review of specification design, coding, purpose of product testing is to verify and validate various work products viz unit integrated unit, final product to ensure that they meet their requirements.

Testing objectives

Basically testing is done for the following purposes.

* Testing is a process of executing program with the intent of finding an error.
* A good test case is one that has a high probability of finding an as yet undiscovered error.
* A successful test case is one that uncovers an as yet undiscovered error.

Our objective is to design test cases that systematically uncover different classes of error and do so with a minimum amount of time and effort. This process has two parts:

* + Planning

This involves writing and reviewing unit integration, functional, validation and acceptance test plans.

* + Execution

This involves executing these test plans, measuring. Collecting data and very fine if it meets the quality criteria. Data collected is used to make appropriate changes in the plans related to development and testing.

The quality of a product or item can be achieved by ensuring that the product meets the requirements by planning and conducting the following tests at various stages.

### Types of Testing Software

The main types of software testing are:

Component Testing

Starting from the bottom the first test level is “Component Testing”, sometimes called Unit Testing. It involves checking that each feature specified in the “Component Design” has been implemented in the component. In theory an independent tester should do this, but in practice the developer usually does it, as they are the only people who understand how a component works. The problem with a component is that it performs only a small part of the functionality of a system, and it relies on co-operating with other parts of the system, which may not have been built yet. To overcome this, the developer either builds, or uses special software to trick the component into believing it is working in a fully functional system.

Interface Testing

As the components are constructed and tested they are then linked together to check if they work with each other. It is fact that two components that have passed all their tests, when connected to each other produce one new component full of faults. These tests can be done by specialists, or by the developers.

Interface testing is not focused on what the components are doing but on how they communicate with each other, as specified in the “System Design”. The “system Design” defines relationship between components, and this involves stating:

1). What a component can expect from another component in terms of services.

2). How these services will be asked for.

3). How they will be given.

4). How to handle non standard conditions, i.e. errors.

5). Tests are constructed to deal with each of these.

The tests are organized to check all the interfaces, until all the components have been built and interfaced to each other producing the whole system.

System Testing

Once the entire system has been built then it has to be tested against the “System Specification” to check if it delivers the features required. It is still developer focused, although specialist developers known as system testers are normally employed to do it.

In essence System testing is not about checking the individual parts of the design, but about checking the system as a whole. In effect it is one giant component.

System testing can involve a number of specialist types of test to see if all the functional and non-functional requirements have been met. In addition to functional requirements these may include the following types of testing for the non-functional requirements:

1). Performance- Are the performance criteria met?

2). Volume- Can large volumes of information be handled?

3). Stress- Can peak volumes of information be handled?

4). Documentation- Is the documentation usable for the system?

5). Robustness- Does the system remain stable under adverse circumstances?

There are many others, the needs for which are dictated by how the system is supposed to perform.

Acceptance Testing

Acceptance testing checks the system against the “Requirements”. It is similar to system testing in that the whole system is checked but the important difference is the change in focus:

System testing checks that the system that was specified has been delivered.

Acceptance testing checks that the system delivers what was requested. The customer and not the developer should always do acceptance testing. The customer knows what is required from the system to achieve value in the business and is the only person qualified to make that judgment. The forms of tests may follow those in system testing, but at all times they are informed by the business needs.

Release Testing

Even if a system meets all its requirements, there is still a case to be answered that it will benefit the business. Release testing is about seeing if the new or changed system will work in the existing business environment. Mainly this means the technical environment, and checks concerns such as:

1). Does it affect any other systems running on the hardware?

2). Is it compatible with other system?

3). Does it have acceptable performance under load?

These tests are usually run by the computer operations team in a business. It would appear obvious that the operation team should be involved right from the start of a project to give their opinion of a new system may have.

Test Case Design

Test case design focuses on a set of techniques for the creation of test cases that meet overall testing objectives. In test case design phase, the engineer creates a series of test cases that are intended to “demolish” the software that has been built.

Any software product can be tested in one of two ways:

1). Knowing the specific function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operational, at the same time searching for errors in each function. This approach is known as Black Box Testing.

2). Knowing the internal workings of a product, tests can be conducted to ensure that internal operation performs according to specifications and all internal components have been adequately exercised. This approach is known as White Box Testing.

Black box testing is designed to uncover errors. They are used to demonstrate that software functions are operations; that input is properly accepted and output is correctly produced; and that integrity of external information is maintained. A black box examines some fundamental aspects of a system with little regard for the internal logical structure of the software.

White box testing of software is predicated on close examination of procedural details. Providing test cases that exercises specific set of conditions and/or loops tests logical paths through the software. The “state of program” may be examined at various points to determine if the expected or asserted status corresponding to the actual status.

Testing of Current Project

Software testing is arguably the least understood part of the development process. Through a four-phase approach, the author shows why eliminating bugs is tricky and why testing is a constant trade-off.

Virtually all developers know the frustration of having software bugs reported by users. When this happens, developers inevitably ask: How did those bugs escape testing? Countless hours doubtless went into the careful testing of hundreds or thousands of variables and code statements, so how could a bug have eluded such vigilance?

Four phases:

* Modeling the software’s environment
* Selecting test scenarios
* Running and evaluating test scenarios
* Measuring testing progress

These phases offer testers a structure in which to group related problems that they must solve before owing on to the next phase.



Performance&Scalability:  
 The scalability in the testing lab features powerful workstations and top-of-the-line networking equipment and is capable of testing the performance and scalability of an end-to-end solution, not just individual components. The lab is capable of generating a realistic load of thousands of virtual users consisting of a mix of user profiles while simultaneously exercising different portions of a dynamic Web application, based on real-world Web usage patterns.  
  
Testing tools can capture a variety of critical statistics, such as per-page user response times, web interactions per second, transaction error rates and network traffic. The testing lab also reports on empirical performance metrics of e-commerce products subjected to massive user load, versus projecting results based on smaller load tests. This approach ensures that the data reported provides an accurate depiction of the actual performance of the tested products. Through arrangements with many of the leading web testing tool vendors, Testing lab is capable of generating complex scripts to be used in testing.  
A typical scalability testing involves the following steps:

* Consult with the client to determine the activities that will be exercised during testing;
* Code the test tool script to stress the website;
* Coordinate the enabling of appropriate server log files during the test;
* Consult with the client regarding a response time threshold value;
* Determine how many concurrent virtual users cause the response time to exceed the threshold value; and
* Calculate average response times under various load conditions.

CompatibilityTesting  
 Testing lab examines a website’s compatibility with multiple operating system/browser combinations. Testing lab works closely with clients to develop a compatibility-testing matrix that will include the relevant combinations to be tested. Available configurations typically include Intel-based systems with Windows operating systems and associated browsers as well as Macintosh systems and associated browsers. Testing lab will test the major features of the website during the compatibility testing, using various approaches.  
Testing lab works with clients to develop a test matrix that reflects the hardware and software combinations to be tested and uses a variety of different editions and service packs of operating environments in the testing. Our Testing lab has identified more than 70 possible combinations of configurations. During testing, Testing labs evaluates the ability of the website to work in a satisfactory manner using both high- and low-speed connections. All major functions of the website will be exercised by each configuration.

FunctionalityTesting  
  
 Functionality in testing labs includes examination of website pages, menus, options, and links to determine if the site functions as intended. Testing labs logs on to a client site and, acting as multiple types of users, performs many functions associated with that site. Testing labs should accomplishes functionality testing by utilizing labs-developed test scripts, client-developed test scripts, or some combination of both. These testing labs works with clients to determine the appropriate operating system and browser combinations to be

tested. Typically, Testing labs exercises all site functions including links, search operations and responsiveness.

UsabilityTesting  
  
 To conduct usability testing, Testing labs assembles a focus group of testers with a broad range of experience in personal computer and Internet usage. Working with its client, all testing lab’s highly skilled Usability Testing Team will develop a questionnaire that addresses the various aspects of the client’s website. The questionnaire, along with the focus group debriefing, is designed to elicit the responses and opinions of the members of the focus group regarding the usability of the website. A final report will include testing labs’s expert opinion as well as the individual responses and group consensus regarding the comparative usability of the website.

Security Implementation

Computer security is an important topic. As e-commerce blossoms, and the Internet works its way into every nook and cranny of our lives, security and privacy come to play an essential role. Computer security is moving beyond the realm of the technical elite, and is beginning to have a real impact on our everyday lives.

It is no big surprise, then, that security seems to be popping up everywhere, from headline news to TV talk shows. Because the general public doesn't know very much about security, a majority of the words devoted to computer security cover basic technology issues such as what firewalls are, what cryptography is, or which antivirus product is best. Much of the rest of computer security coverage centers around the "hot topic of the day," usually involving an out-of-control virus or a malicious attack. Historically, the popular press pays much attention to viruses and denial-of-service attacks: Many people remember hearing about the Anna Kournikova worm, the "Love Bug," or the Melissa virus ad nauseam. These topics are important, to be sure. Nonetheless, the media generally manages not to get to the heart of the matter when reporting these subjects. Behind every computer security problem and malicious attack lies a common enemy—bad software.Complex systems, by their very nature, introduce multiple risks. And almost all systems that involve software are complex. One risk is that malicious functionality can be added to a system (either during creation or afterward) that extends it past its primary, intended design. As an unfortunate side effect, inherent complexity lets malicious and flawed subsystems remain invisible to unsuspecting users until it is too late. This is one of the root causes of the malicious code problem. Another risk more relevant to our purposes is that the complexity of a system makes it hard to understand, hard to analyze, and hard to

secure. Security is difficult to get right even in simple systems; complex systems serve only to make security harder. Security risks can remain hidden in the jungle of complexity, not coming to light until it is too late.

Extensible systems, including computers, are particularly susceptible to complexity-driven hidden risk and malicious functionality problems. When extending a system is as easy as writing and installing a program, the risk of intentional introduction of malicious behavior increases drastically—as does the risk of introducing unintentional vulnerabilities.

Any computing system is susceptible to hidden risk. Rogue programmers can modify systems software that is initially installed on the machine. Unwitting programmers may introduce a security vulnerability when adding important features to a network-based application. Users may incorrectly install a program that introduces unacceptable risk or, worse yet, accidentally propagate a virus by installing new programs or software updates. In a multiuser system, a hostile user may install a Trojan horse to collect other users' passwords. These attack classes have been well-known since the dawn of computing, so why is software security a bigger problem now than in the past? We believe that a small number of trends have a large amount of influence on the software security problem.

One significant problem is the fact that computer networks are becoming ubiquitous. The growing connectivity of computers through the Internet has increased both the number of attack vectors (avenues for attack) and the ease with which an attack can be made. More and more computers, ranging from home personal computers (PCs) to systems that control critical infrastructures (such as the power grid), are being connected to the Internet. Furthermore, people, businesses, and governments are increasingly dependent on network-enabled communication such as e-mail or Web pages provided by information systems. Unfortunately, because these systems are connected to the Internet, they become vulnerable to attacks from distant sources. Put simply, an attacker no longer needs physical access to a system to cause security problems.

Because access through a network does not require human intervention, launching automated attacks from the comfort of your living room is relatively easy. Indeed, the well-publicized denial-of-service attacks in February 2000 took advantage of a number of (previously compromised) hosts to flood popular e-commerce Web sites, including Yahoo!, with bogus requests automatically. The ubiquity of networking means that there are more systems to attack, more attacks, and greater risks from poor software security practice than ever before.

A second trend that has allowed software security vulnerabilities to flourish is the size and complexity of modern information systems and their corresponding programs. A desktop system running Windows/NT and associated applications depends on the proper functioning of the kernel as well as the applications to ensure that an attacker cannot corrupt the system. However, NT itself consists of approximately 35 million lines of code, and applications are becoming equally, if not more, complex. When systems become this large, bugs cannot be avoided.

Exacerbating this problem is the widespread use of low-level programming languages, such as C or C++, that do not protect against simple kinds of attacks (most notably, buffer overflows). However, even if the systems and applications codes were bug free, improper configuration by retailers, administrators, or users can open the door to attackers. In addition to providing more avenues for attack, complex systems make it easier to hide or to mask malicious code. In theory, we could analyze and prove that a small program was free of security problems, but this task is impossible for even the simplest of desktop systems today, much less the enterprise-wide systems used by businesses or governments.

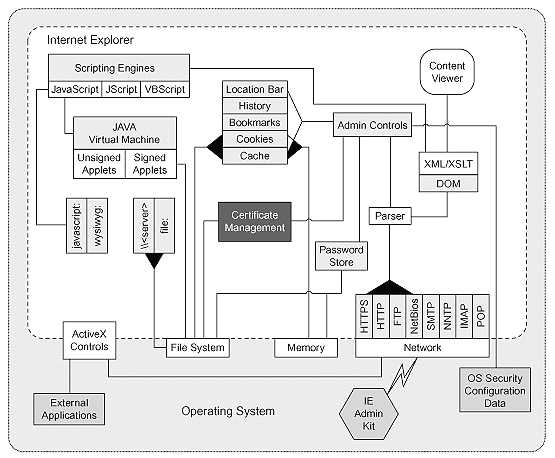
A third trend exacerbating software security problems is the degree to which systems have become extensible. An extensible host accepts updates or extensions, sometimes referred to as **mobile code**, so that the system's functionality can be evolved in an incremental fashion. For example, the plug-in architecture of Web browsers makes it easy to install viewer extensions for new document types as needed.

Given a basic intuitive grasp of the architecture of a browser (a program that runs on top of an operating system and provides basic Web interface services), it is natural to assume that a browser may be used to enhance security. In reality, it is hard to tell where the boundaries of the browser are, where the operating system fits, and how a browser can protect itself. The two most popular browsers, Netscape Navigator and Microsoft Internet Explorer (MSIE), have very fuzzy boundaries and include many more hooks to external applications than most people realize.

On the most pervasive platform (Windows 95, 98, and Millennium Edition (ME)) there is really no way for a browser to protect itself or any secrets it may be trying to keep (like client-side certificates) at all. This means that if your design requires some security features inside the browser (like an intact Java Virtual Machine [ JVM] or a cryptographic secret), there is probably a real need for a more advanced operating system like Windows/NT or UNIX.

Without delving into the details of how a browser is constructed, it is worth showing a general-purpose abstract architectural Netscape and MSIE respectively. From a high-level perspective, it is clear that there are many interacting components involved in each architecture. This makes securing a browser quite a monumental task. In addition, helper applications (such as AOL Instant Messenger for Netscape and ActiveX control functionality in MSIE) introduce large security risks.Unfortunately, the very nature of extensible systems makes security harder. For one thing, it is hard to prevent malicious code from slipping in as an unwanted extension. Meaning, the features designed to add extensibility to a system (such as Java's class-loading mechanism) must be designed with security in mind. Furthermore, analyzing the security of an extensible system is much harder than analyzing a complete system that can't be changed. How can you take a look at code that has yet to arrive? Better yet, how can you even begin to anticipate every kind of mobile code that may arrive?

**Internet Explorer Architecture**

****

Together, the three trends of ubiquitous networking, growing system complexity, and built-in extensibility make the software security problem more urgent than ever. There are other trends that have an impact as well, such as the lack of diversity in popular computing environments, and the tendency for people to use systems in unintended ways (for example, using Windows NT as an embedded operating system).

Software Maintenance

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

|  |  |  |
| --- | --- | --- |
| The Software Maintenance Context | | |
| Existing software product | Software Maintenance  → | Modified software product |
| Maintenance need   * Enhancement * Correction | Software engineering process | Installation, operation, maintenance |

|  |  |  |  |
| --- | --- | --- | --- |
| Software Maintenance Covers | | | |
| Basic concepts | Maintenance Process | Key issues in software maintenance | Techniques for maintenance |
| * Definitions and terminology * Majority of maintenance costs * The nature of maintenance * Evolution of software * Need for maintenance * Categories of maintenance | * Process models * Maintenance activities | * Technical * Management * Cost and estimation * Software maintenance measurement | * Program comprehension * Re-engineering * Reverse engineering * Impact analysis |

FUTURE SCOPE

The Project “*Shopping Hub – An Online Store*” is generalized software and can be easily used in any online shopping website with little or no change. The changes in software can be easily accommodated. The addition and deletion of the modules in software can be easily adjusted. The software can be enhanced up to any legal extent depending upon user’s requirement. It will be able to serve the organization even if it increases its services. I have completed the project successfully and according to the developed project fully satisfied their needs except certain things because everything is not perfect in this materialistic world and due to certain factors, which directly or indirectly affect the project. We hope that this project will serve its purpose for which it is developed there by underlining success of our project.

Up till now all the activities of the organization are performed manually, it requires more human efforts to manage the each and every aspects of organization which is very costly process. To overcome all kind of problems being faced by organization our project “online-Shopping Computer Hardware” will provide efficient and cost effective solution and always have possibilities of enhancement up any legal extent to satisfy user requirement. This system provides a greater solution for those who are less interacted with the computer system because it provides the graphical user interface facility. This software will help to shopping with wasting your valuable time in crowded malls and large traffic jams.

LIMITATION OF THE PROJECT

There are several points in this project that have not been addressed.

Some of these are as follows:

1. There is lack of system security, reliability, standards and some communication protocols.
2. It is used in business Purpose .
3. A wide variety of tool& technologies as involved in this project that makes it cumbersome.
4. There arno Registration and hosting of website .
5. The project contained in this document can be viewed only on Microsoft intetnet explorer only it is hosted.
6. Software development tools are still evolving and changing rapidly.
7. Vendors may need special web services and other infrastructure, in addition to the network severs.
8. Cost is high.
9. Lack of trust and user resistance.
10. Lack of touch and feel online.

Bibliography

Websites

|  |  |
| --- | --- |
| URL’s | .NET Resource Description |
| [www.gotdotnet.com](http://www.gotdotnet.com) | All about asp.net |
| [www.123aspx.com](http://www.123aspx.com) | Directory of .NET articles |
| [www.vbws.com](http://www.vbws.com) | Building XML web services with VB. Net |
| [www.soapwebservices.com](http://www.soapwebservices.com) | SOAP Web Services |
| [www.w3.org](http://www.w3.org) | HTML & XML Specification |
| [www.xmlws101.com](http://www.xmlws101.com) | XML web services for ASP.NET |
| [www.mindtools.com](http://www.mindtools.com) | For SAD & Software Engg tools |
| [www.xmlwebservices.com](http://www.xmlwebservices.com) <http://msdn.microsoft.com/> | XML Web Services |

Books

|  |  |
| --- | --- |
| Name of Book | Publication |
| Developing web applications with Vb.NET and ASP.NET | Wiley |
| .NET E-Commerce programming | Sybex |
| Mastering ASP.NET | BPB |
| Complete Rereference ASP.NET | TMH |
| Visual Studio.NET programming | Wiley |
| Visual Basic.Net Programming Black Book | Wiley Dreamtech |