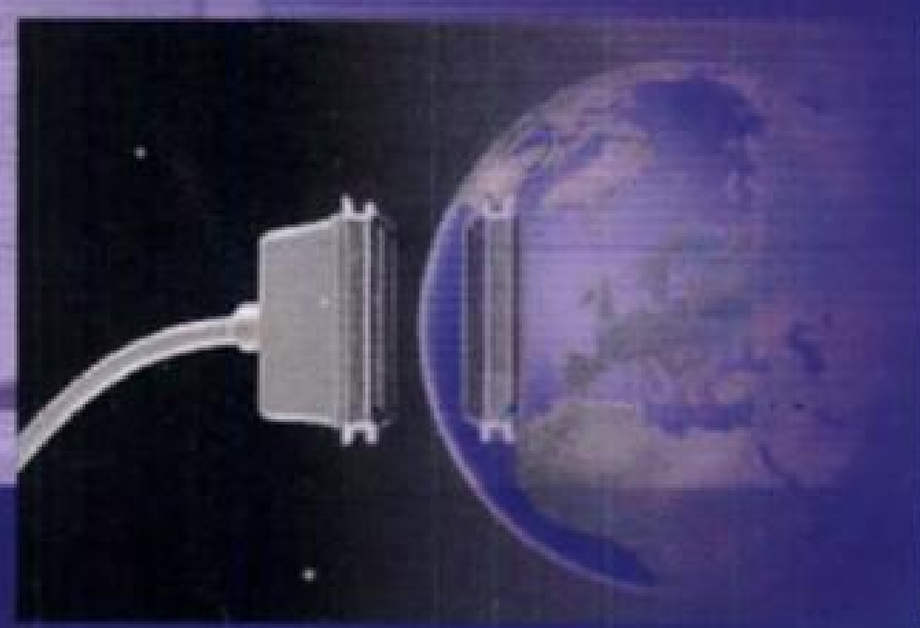
WEB TECHNOLOGY & DESIGN

INCLUDING

- *HTML*
- *JavaScript*
- *JSP/ASP*
- *JDBC*
- *Servlets*
- *XML*

AND PROJECTS

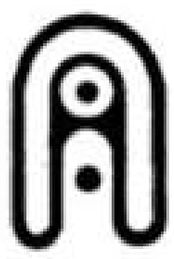


C. Xavier

NEW AGE INTERNATIONAL PUBLISHERS

WEB TECHNOLOGY & DESIGN

C. Xavier



NEW AGE

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS

New Delhi • Bangalore • Chennai • Cochin • Guwahati • Hyderabad

Jalandhar • Kolkata • Lucknow • Mumbai • Ranchi

First Edition : 2003
Reprint : 2005

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS
4835/24, Ansari Road, Daryaganj,
New Delhi - 110 002
Visit us at : www.newagepublishers.com

Offices at :
Bangalore, Chennai, Cochin, Guwahati, Hyderabad, Jalandhar,
Kolkata, Lucknow, Mumbai and Ranchi

Rs. 220.00

ISBN : 81-224-1450-8

3 4 5 6 7 8 9 10

Published by New Age International (P) Ltd.,
4835/24, Ansari Road, Daryaganj, New Delhi-110 002 and
printed in India at Chaman Offset, New Delhi-110 002

Contents

CHAPTER 1

Internet Principles

In 1984 *William Gibson* in his sci-fi novel *Neuromancer* coined the term *Cyberspace*. It refers to the non-physical terrain created by computer systems. Online systems, for example, create a *cyberspace* within which people can communicate with one another (via e-mail), do research, or simply window shop. Like physical space, cyberspace contains *objects* (files, mail messages, graphics, etc.) and different modes of transportation and delivery. Unlike real space, exploring cyberspace does not require any physical movement other than pressing keys on a keyboard or moving a mouse. Some programs, particularly computer games, are designed to create a special cyberspace, one that resembles physical reality in some ways but defies it in others. In its extreme form, called *virtual reality*, users are presented with visual, auditory, and even tactile feedback that makes cyberspace feel real. The dreams of *William Gibson* is becoming a reality in the Internet world.

1.1 Introduction to Internet

A *network of computers* refers to a group of computers connected with each other as per a topology. The computers in a network are capable of sharing the resources among themselves, the resources such as memory, peripherals, etc. They communicate with each other in a defined way. Internet refers to a global network connecting millions of computers. More than 100 countries are linked into exchanges of data, news and opinions. Unlike *online network services*, which are centrally controlled, the Internet is decentralized by design. Each Internet computer, called a *host*, is independent. Its operators can choose which Internet services to use and which local services to make available to the global Internet community. Remarkably, this anarchy by design works exceedingly well. There are a variety of ways to access the Internet. Most of the Internet users gain access through a commercial *Internet Service Provider* (ISP) as illustrated in Figure 1.1. Usually we connect our Home PC to the Internet through the landline telephone connection

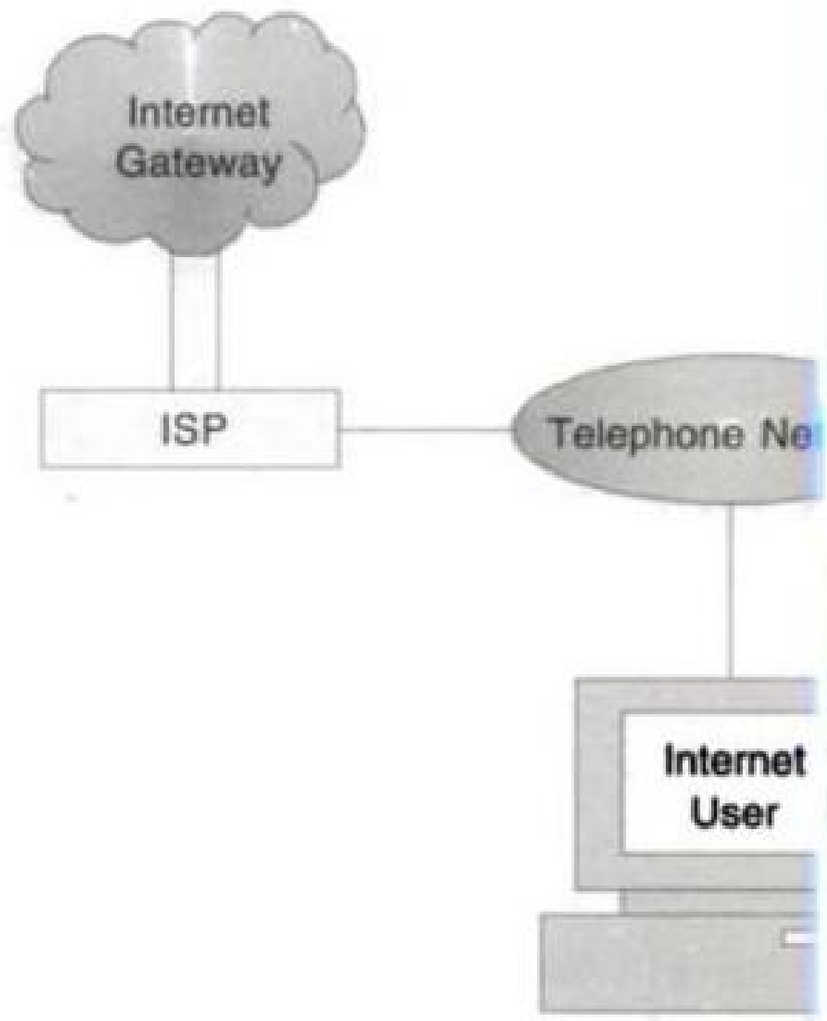
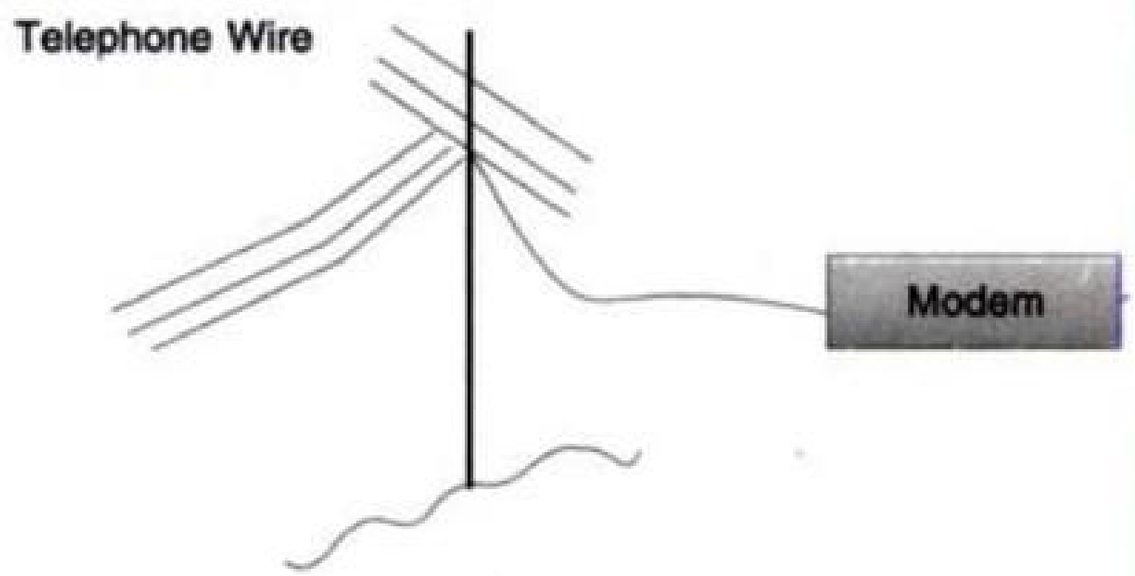


Figure 1.1 Connecting throu

using a device called modem as shown in Figure 1.2. So, t g two factors decide the characteristics of the Internet services.

- Modem
- Connectivity for communication



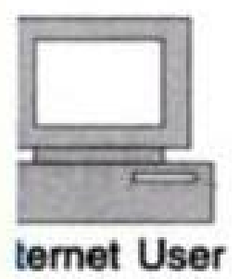


Figure 1.2 Internet Connec

1.1.1 Modem

A modem (short form of ***modulator-demodulator)*** is a devi les a computer to trans- mit data over, for example, telephone or cable lines. Comp ation is stored digitally, whereas information transmitted over telephone lines is tran e form of analog waves. A modem converts between these two forms.

Fortunately, there is one standard interface for connecting external modems to computers called *RS-232*. Consequently, any external modem can be attached to any computer that has an RS-232 port, which almost all personal computers have. There are also modems that come as an expansion board that you can insert into a vacant expansion slot. These are sometimes called *onboard* or *internal modems*.

While the modem interfaces are standardized, a number of different protocols for formatting data to be transmitted over telephone lines exist. Some, like CCITT V.34, are official standards, while others have been developed by private companies. Most modems have built-in support for the more common protocols—at slow data transmission speeds at least, most modems can communicate with each other. At high transmission speeds, however, the protocols are less standardized.

Characteristics of a Modem

Aside from the transmission protocols that they support, the following characteristics distinguish one modem from another:

1. Speed of transmission
At slow rates, modems are measured in terms of baud rates. The slowest rate is 300 baud (about 25 characters per second—cps). At higher speeds, modems are measured in terms of bits per second (*bps*). A fast modem transmits at 57,600 bps, although they can achieve even higher data transfer rates by compressing the data. Obviously, the faster the transmission rate, the faster you can send and receive data. Note, however, that you cannot receive data any faster than it is being sent. If, for example, the device sending data to your computer is sending it at 2,400 bps, you must receive it at 2,400 bps. It does not always pay, therefore, to have a very fast modem. In addition, some telephone lines are unable to transmit data reliably at very high rates.

2. Modes supported
Many modems support a switch to change between voice and data modes. In data mode, the modem acts like a regular modem. In voice mode, the modem acts like a regular telephone. Modems that support a voice/data switch have a built-in loudspeaker and microphone for voice communication.

3. Auto-answer
An auto-answer modem enables your computer to receive calls in your absence. This is only necessary if you are offering some type of computer service that people can call in to use.

4. Data compression
Some modems perform data compression, which enables them to send data at faster rates. However, the modem at the receiving end must be able to decompress the data using the same compression technique.

5. Flash memory

Some modems come with *flash memory* rather than conventional ROM, which means that the communication protocols can be easily updated if necessary.

6. Fax capability

Most modern modems are fax modems, which means that they can send and receive faxes.

1.1.2 Connectivity for Communication

Connectivity is considered to be the backbone of the Internet world. Apart from ordinary telephone line we are also using broad band connectivity such as ISDN connections.

1.1.3 Standards of Data Communication

The *CCITT (Comité Consultatif International Téléphonique et Télégraphique*) is an organization that sets international communications standards. It has developed the X.400 standard, which attempts to provide a universal way of addressing messages. To date, though, the de facto addressing standard is the one used by the Internet system because almost all e-mail systems have an Internet gateway.

CCITT has defined many important standards for data communications, including the following:

- **Group 3:** The universal protocol for sending fax documents across telephone lines. The Group 3 protocol specifies *CCITT T.4 data compression* and a maximum transmission rate of 9,600 baud. There are two levels of resolution: 203 by 98 and 203 by 196.
- **Group 4:** A protocol for sending fax documents over ISDN networks. The Group 400 protocol supports images of up to 400 dpi (dots per inch) resolution.
- **V.21:** The standard for full-duplex communication at 300 baud in Japan and Europe. In the United States, Bell 103 is used in place of V.21.
- **V.22:** The standard for half-duplex communication at 1,200 bps in Japan and Europe. In the United States, the protocol defined by Bell 212A is more common.
- ***V.22bis*:** The worldwide standard for full-duplex modems sending and receiving data across telephone lines at 1,200 or 2,400 bps.
- **V.29:** The standard for half-duplex modems sending and receiving data across telephone lines at 1,200, 2,400, 4,800, or 9,600 bps. This is the protocol used by fax modems.
- **V.32:** The standard for full-duplex modems sending and receiving data across phone lines at 4,800 or 9,600 bps. V.32 modems automatically adjust their transmission speeds based on the quality of the lines.

The original version of ISDN employs baseband transmission. Another version, called B-ISDN, uses broadband transmission and is able to support transmission rates of 1.5 MBPS. B-ISDN requires fiber optic cables and is not widely available at present. The baseband and broadband transmission are discussed in section 1.1.6.

1.1.5 DSL (Digital Subscriber Lines) Connection

DSL is the acronym for D*igital Subscriber Lines.* DSL technologies use sophisticated modulation schemes to pack data onto copper wires. They are sometimes referred to as *last-mile technologies* because they are used only for connections from a telephone switching station to a home or office, not between switching stations. DSL is similar to ISDN in as much as both operate over existing copper telephone lines and both require the short runs to a central telephone office (usually less than 20,000 feet). However, DSL offers much higher speeds—up to 32 Mbps for upstream traffic (client to server) , and from 32 Kbps to over 1 Mbps for downstream traffic (server to client).

There two main categories being

- Asymmetric digital subscriber line (ADSL)
- Symmetric digital subscriber line (SDSL)

Depending upon the speed some also classify as follows:

- High-data-rate DSL (HDSL)
- Very high DSL (VDSL).

Asymmetric Digital Subscriber Line (ADSL)

Asymmetric digital subscriber line is a new technology that allows more data to be sent over existing copper telephone lines. ADSL supports data transfer downstream rate from 1.5 to 9 Mbps and upstream rate from 16 to 640 Kbps. ADSL requires a special ADSL modem. ADSL is growing in popularity as more areas around the world gain access to Internet.

Symmetric Digital Subscriber Line (SDSL)

Symmetric digital subscriber line is a technology that allows more data to be sent over existing copper telephone lines. SDSL supports data rates up to 3 Mbps. SDSL works by sending digital pulses in the high-frequency area of telephone wires. Since these high frequencies are not used by normal voice communications, SDSL can operate simultaneously with voice connections over the same wires. SDSL requires a special SDSL modem. SDSL is called symmetric because it supports the same data rates for upstream and downstream traffic. ADSL is more popular in North America, whereas SDSL is being developed primarily in Europe. A comparison of ADSL and SDSL is shown in Table 1.1.2.

Table 1.1.2 DSL Type

DSL Type	*Upstream Speed*	*Downstream Speed*	*Popularity*
Asymmetric	16 to 6 to kbps	1.5 to 9 Mbps	North America
Symmetric	upto 3 Mps	upto 3 Mbps	Europe

1.1.6 Broadband and Baseband Transmissions

Broadband transmission is a type of data transmission in which a single medium (wire) can carry several channels at once. Cable TV, for example, uses broadband transmission. In contrast, baseband transmission allows only one signal at a time.

Most communications between computers, including the majority of local-area networks use only baseband communications. An exception is B-ISDN network., which employ broadband transmission.

1.2 Client Server Model

Internet works in a client server model. This section throws some light on the servers that are used in the marketplace today. Server Platforms refers to the *operating system* that drives the server.

1.2.1 Application Servers

Application Servers are the type of middleware, which occupy a large chunk of computing territory between database servers and the end user, and they often connect the two.

1.2.2 Audio/Video Servers

Audio/Video Servers bring multimedia capabilities to Web sites by enabling them to broadcast streaming multimedia content.

1.2.3 Chat Servers

Chat Servers enable a large number of users to exchange information in an environment similar to Internet newsgroups that offer real-time discussion capabilities.

1.2.4 Fax Servers

Fax Servers are ideal solution for organizations looking to reduce incoming and outgoing telephone resources but that need to fax actual documents.

1.2.5 FTP Servers

FTP Servers serve one of the oldest Internet services. It makes it possible to move one or more files securely between computers while providing file security and organization as well as transfer control.

1.2.6 Groupware Servers

A *groupware server* is a software designed to enable users to collaborate, regardless of location, via the Internet or a corporate intranet and to work together in a virtual atmosphere.

1.2.7 IRC Servers

IRC Servers provide an option for those seeking real-time discussion capabilities, Internet Relay Chat consists of various separate networks (or "nets") of servers that allow users to connect to each other via an IRC network.

1.2.8 List Servers

List servers offer a way to better manage mailing lists, whether they be interactive discussions open to the public or one-way lists that deliver announcements, newsletters, or advertising.

1.2.9 Mail Servers

Mail servers move and store mail over corporate networks (via LANs and WANs) and across the Internet.

1.2.10 News Servers

News servers act as a distribution and delivery source for the thousands of public news groups currently accessible over the USENET news network.

1.2.11 Proxy Servers

Proxy servers sit between a client program (typically a Web browser) and an external server (typically another server on the Web) to filter requests, improve performance, and share connections.

1.2.12 Web Servers

A *web server* is a computer system (Hardware and software) that delivers (*serves up*) Web pages. Every Web server has an IP address and possibly a domain name. For example, if you enter the

URL `http://www.stxaviers.com/index.html` in your browser, this sends a request to the server whose domain name is *`stxavier.com`*. The server then fetches the page named *`index.html`* and sends it to your browser.

Any computer can be turned into a Web server by installing server software and connecting the machine to the Internet. There are many Web server software applications, including public domain software from NCSA and Apache, and commercial packages from Microsoft, Netscape and others.

1.3 Protocol

A *protocol* is a program written as per mutually accepted standard that two computers use to communicate with each other. Computers use protocols (protocol programs) to format consistently their messages so that other computers can understand them, acknowledge the receipt of messages, indicate that they are finished sending a message and so on. In the network when one computer requests for the service of another, it is called a client. In order to establish the needed connectivity, both the client machine and the server machine must have a common protocol program. This is illustrated in Figure 1.3.

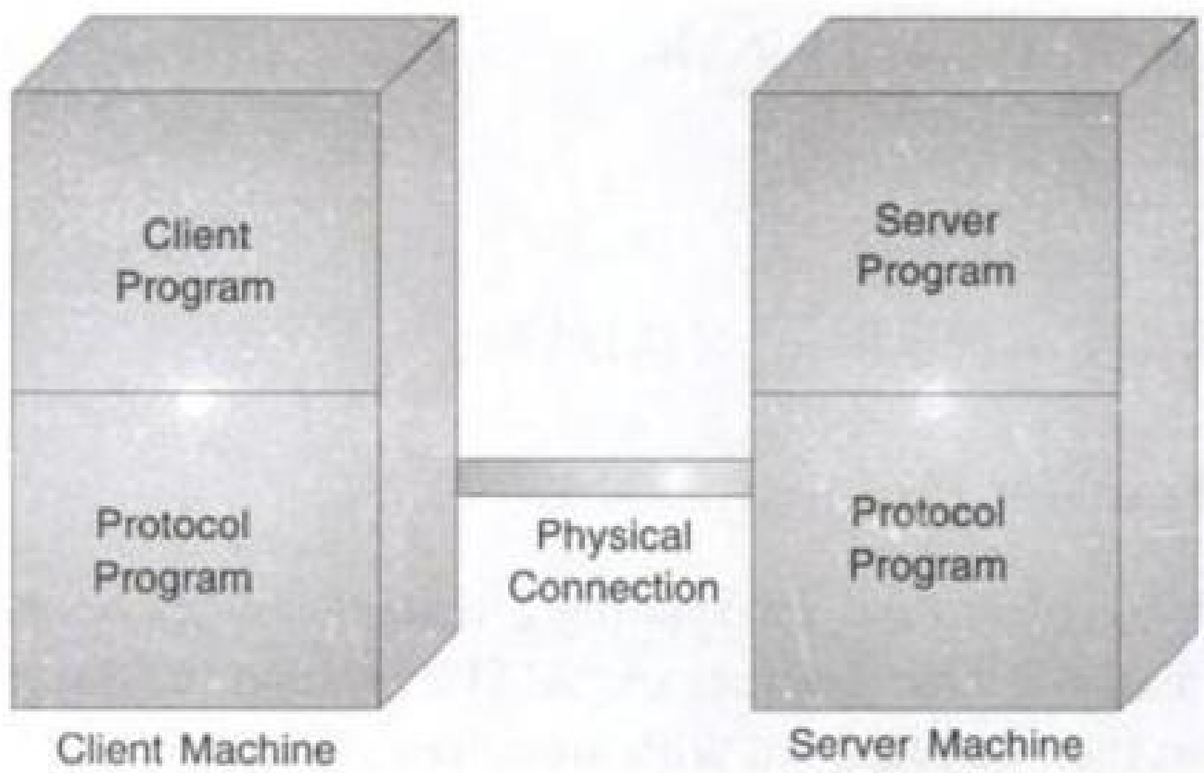


Figure 1.3 The Client-Server Communication

In short a protocol is an agreed-upon format for transmitting data between two devices. The protocol determines the following:

- *the type of error checking to be used*
- *data compression method, if any*
- *how the sending device will indicate that it has finished sending a message*
- *how the receiving device will indicate that it has received a message*

There are a variety of standard protocols from which programmers can choose. Each has particular advantages and disadvantages; for example, some are simpler than others, some are more reliable, and some are faster.

From a user's point of view, the only interesting aspect about protocols is that your computer or device must support the right ones if you want to communicate with other computers. The protocol can be implemented either in hardware or in software.

Some standard protocols used in networks are listed below:

1. Simple Mail Transfer Protocol (SMTP)
2. Post Office Protocol version 3 (POP3)
3. Point to point Protocol/Serial Line Interface Protocol (POP/SLIP)
4. Transmission Control Protocol/Internet Protocol (TCP/IP)
5. Hyper Text Transfer Protocol (HTTP)
6. File Transfer Protocol (FTP)
7. Internet Mail Access Protocol (IMAP)
8. Internet Relay Chat (IRC)
9. Network News Transfer Protocol (NNTP)
10. Telnet
11. Gopher
12. Light weight Directory Access Protocol (LDAP)

1.3.1 Simple Mail Transfer Protocol

Simple Mail Transfer Protocol is used for sending e-mail messages between servers. Most e-mail systems that send mail over the Internet use SMTP to send messages from one server to another; the messages can then be retrieved with an e-mail client using either POP or IMAP. In addition, SMTP is generally used to send messages from a mail client to a mail server. This is why you need to specify both the POP or IMAP server and the SMTP server when you configure your e-mail application.

1.3.2 Post Office Protocol (POP)

Post Office Protocol is a protocol used to retrieve e-mail from a mail server. Most e-mail applications (e-mail client) use the POP protocol, although some can use the newer protocol such as *IMAP (Internet Message Access Protocol).* There are two versions of POP. The first, called *POP2,* became a standard in the mid-80's and requires SMTP to send messages. The newer version, POP3, can be used with or without SMTP.

1.3.3 TCP/IP on the Internet

TCP/IP is the widely accepted protocol used in the networks. It is a protocol suite that consists of several protocols including two primary protocols, Transmission Control Protocol (TCP) and Internet Protocol (IP). The TCP/IP Protocol (protocol program) works asynchronously to handle multiple message traffic simultaneously from multiple sources to multiple destinations as shown in Figure 1.4. TCP (TCP Protocol program) handles the data integrity. It makes sure data gets to the destination without errors. TCP is also responsible for disassembling and assembling the data. It divides large messages into smaller packets of at the most 15 KB size. Each of this packet is called a segment. TCP numbers these segments and hands over to the IP. IP is the Protocol (protocol program) that controls how data moves around on the network. After TCP divides the message into segment, IP labels them with source and destination. These packets are now called *IP Datagrams*. IP also takes care of sending the datagrams by determining the route. These datagrams may hop several networks before reaching the destination.

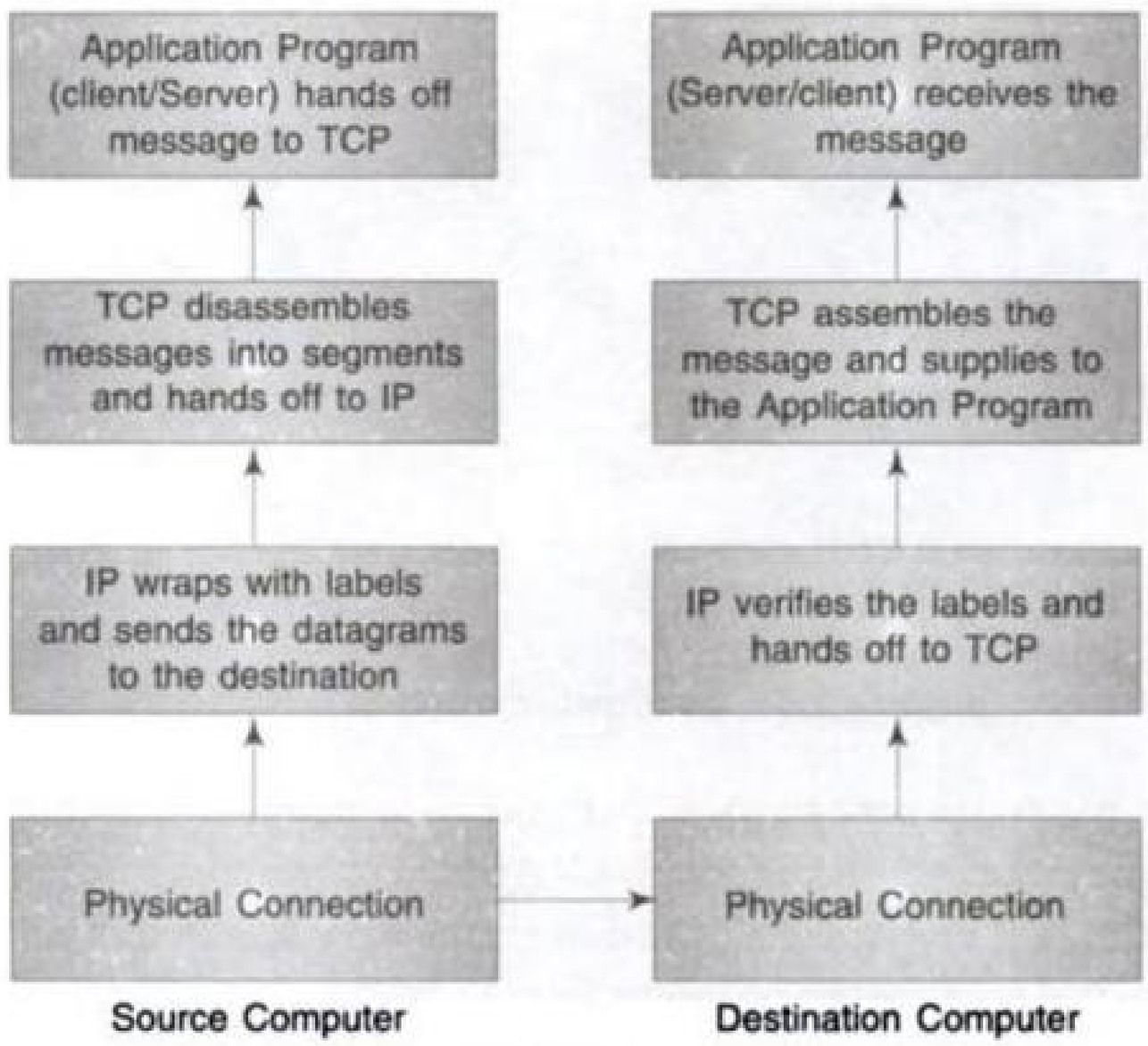


Figure 1.4 TCP/IP Communication

At the destination the IP verifies the labels and passes them to the TCP. The TCP (at the destination) checks if all the segments have been received. If any of the segments are missing it informs this to the source TCP and requests the segment to be sent again. It is called *retry*. After verification, the TCP assembles the message from these data segments and supplies to the destination program.

Packet Switching (Routing)

Routing (Packet Switching) refers to the job of transferring the data packets (IP datagrams) to an appropriate computer. A special computer that does routing is called router. We have already seen that Internet is a network of networks. So when a packet of data (IP datagram) starts from a computer, it is submitted to the router of the network to which the computer belongs to. The router verifies the IP address of the destination. The destination network may not be directly reached. The router finds the next network to which the datagram must be submitted and does it. Similarly the datagram passes several routers and finally reaches the final destination network. Every router uses routing tables and routing algorithms to accomplish the job.

The routing algorithm will choose the next network to which the datagram may be routed depending upon the data traffic and the shortest route. Datagrams of the same message may travel through different routes to reach the final destination. But finally the TCP protocol takes care of verifying if all the datagrams have reached. For example, suppose a Computer C1 in network A wants to communicate with a Computer C2 in network F as shown in Figure 1.5.

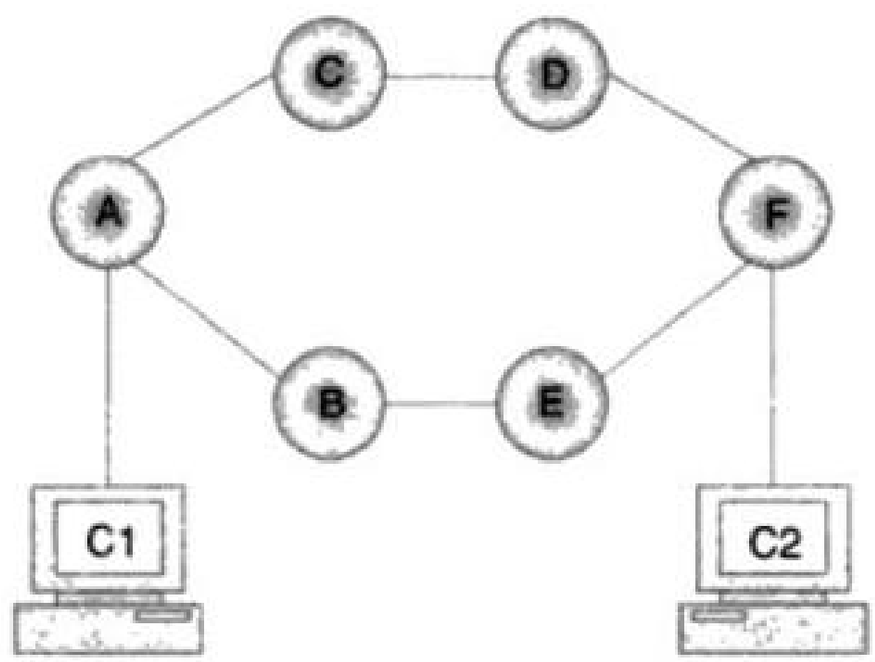


Figure 1.5 Communication in a Network

When a message has to be sent from A to F, the message is disassembled into IP datagrams. Some datagrams may travel in the path C1ACDFC2 whereas some may travel through C1ABEFC2. However, C2 verifies if all the datagrams have reached. Then it assembles the message and supplies to the application program in C2.

Tracing the Route (Tracert Command)

In Windows or Windows NT when we are connected to Internet we can see the route from our computer to any Internet host. This is got by the tracert command in command prompt. In Windows NT, first open the DOS cmd prompt by running cmd command. In the DOS cmd prompt type

```
tracert<domain>
```

For example, we can type tracert lsu.edu

Router

A router is a special computer that manages the traffic from network to network. The router determines the path of travel for a datagram in TCP/IP. Routers keep track of the next computer to which the datagram has to hop. They use routing tables and routing algorithms to do routing.

In Windows NT, we can see the IP address of the router in our network by running the WINIPCFG.EXE program. When we run the program click more info and then select PPP Adapter in the dropdown list. In the middle of the dialog box the IP address of the router is shown in the field Default Gateway. This is the IP address of the router to which your computer will send a TCP/IP datagram when it is addressed to a computer that is not on your local network. This is illustrated in Figure 1.6.

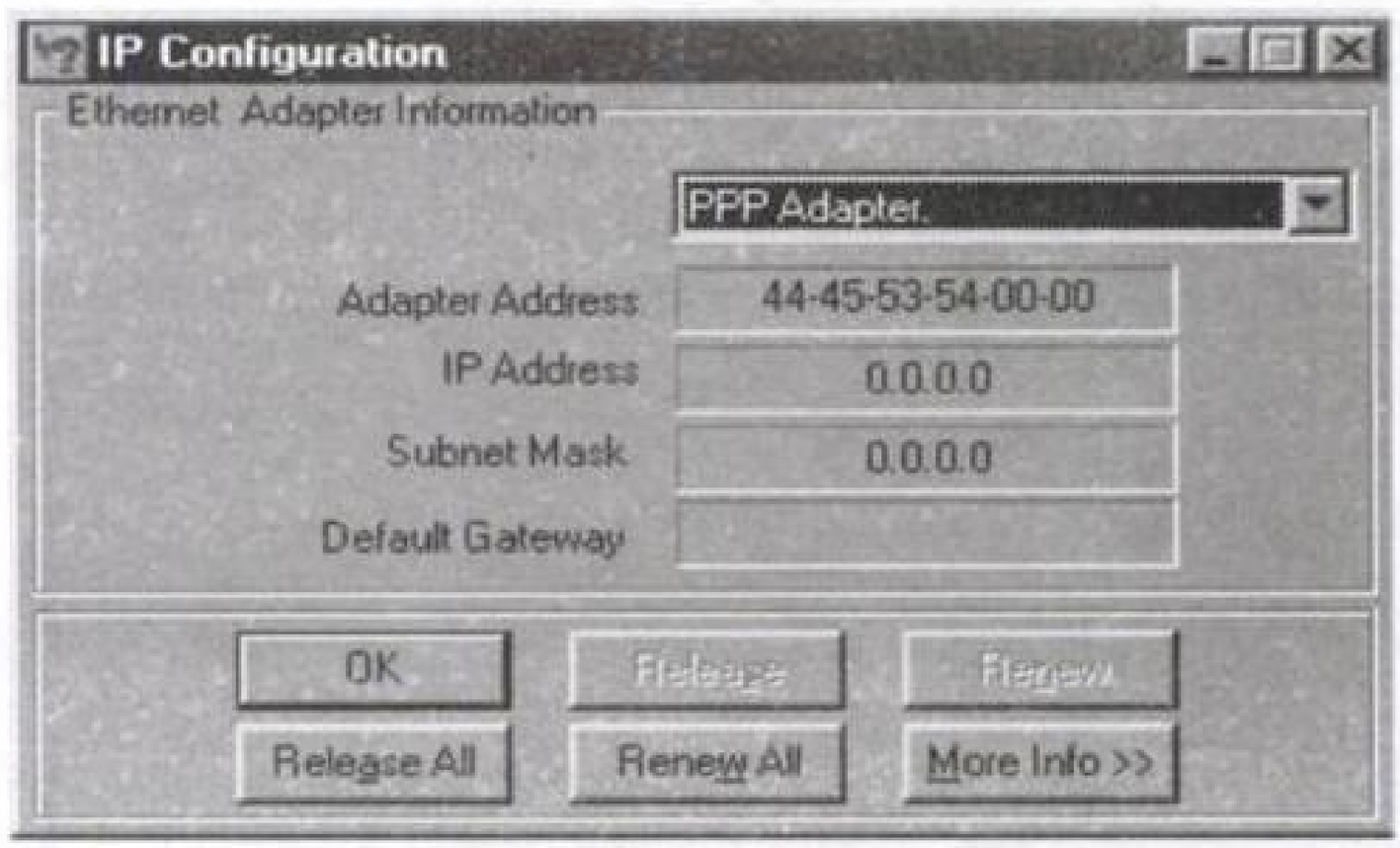


Figure 1.6 IP Address of Router

1.3.4 HyperText Transfer Protocol (HTTP)

HyperText Transfer Protocol is the underlying protocol used by the *World Wide Web*. HTTP defines how messages are formatted and transmitted, and what action Web servers and browsers should take in response to various commands. For example, when you enter a URL in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested Web page. The other main standard that controls how the World Wide Web works is HTML, which covers how Web pages are formatted and displayed.

HTTP is called a *stateless* protocol because each command is executed independently, without any knowledge of the commands that came before it. This is the main reason that it is difficult to implement Web sites that react intelligently to user input. This shortcoming of HTTP is being addressed in a number of new technologies, including *ActiveX, Java, JavaScript* and *cookies*.

1.3.5 File Transfer Protocol (FTP)

File Transfer Protocol or FTP, is a protocol used to upload files from a workstation to a FTP server or download files from a FTP server to a workstation. It is the way that files get transferred from one device to another in order for the files to be available on the Internet. When `ftp` appears in a URL it means that the user is connecting to a file server and not a Web server and that some form of file transfer is going to take place. Most FTP servers require the user to log on to the server in order to transfer files.

In contrast, Hyper Text Transfer Protocol, or HTTP, is a protocol used to transfer files from a Web server onto a browser in order to view a Web page that is on the Internet. Unlike FTP, where entire files are transferred from one device to another and copied into memory, HTTP only transfers the contents of a web page into a browser for viewing. FTP is a two-way system as files are transferred back and forth between server and workstation. HTTP is a one-way system as files are transported only from the server onto the workstation's browser. When `http` appears in a URL it means that the user is connecting to a Web server and not a file server. The files are transferred but not downloaded, therefore not copied into the memory of the receiving device.

1.3.6 Wireless Application Protocol (WAP)

Wireless Application Protocol defines a secure specification that allows users to access Internet information instantly via handheld wireless devices such as mobile phones, pagers, two-way radios, *smartphones* and communicators. WAP supports most wireless networks. WAP is supported by all operating systems. Ones specifically engineered for handheld devices now include PalmOS, EPOC, Windows CE, FLEXOS, OS/9, and JavaOS. WAPs that use displays and access the Internet run using *microbrowsers. Microbrowsers* are browsers with small file sizes that can accommodate the low memory constraints of handheld devices and the low-bandwidth constraints of a wireless-handheld network.

Although WAP supports HTML and XML, the WML language (an XML application) is specifically devised for small screens and one-hand navigation without a keyboard. WML is scalable from two-line text displays up through graphic screens found on items such as smart phones and communicators. WAP also supports WMLScript. It is similar to JavaScript, but makes minimal demands on memory and CPU power because it does not contain many of the unnecessary functions found in other scripting languages. Because WAP is fairly new it is still an initiative that was started by Unwired Planet, Motorola, Nokia, and Ericsson.

.3.7 Bluetooth

Bluetooth refers to a short-range radio technology aimed at simplifying communications among Net devices and between devices and the Internet. It also aims to simplify data synchronization

between Net devices and other computers. Products with Bluetooth technology must be qualified and pass interoperability testing by the Bluetooth Special Interest Group prior to release.

1.3.8 Simple Object Access Protocol (SOAP)

Simple Object Access Protocol proposed by Microsoft provides a way for applications to communicate with each other over the Internet, independent of platform. Unlike IIOP, SOAP piggybacks a DOM onto HTTP (port 80) in order to penetrate server firewalls, which are usually configured to accept port 80 and port 21 (FTP) requests. SOAP relies on XML to define the format of the information and then adds the necessary HTTP headers to send it.

1.3.9 Internet Inter-ORB Protocol (IIOP)

Internet Inter-ORB Protocol is a protocol developed by the *Object Management Group* (OMG) to implement CORBA solutions over the World Wide Web. IIOP enables browsers and servers to exchange integers, arrays, and more complex objects, unlike HTTP, which only supports transmission of text.

1.4 Internet IP Address

In the global network Internet, each node is identified with unique number called IP address. The IP address of a machine is an array of four numbers separated by period, as shown below:

18.10.200.14

185.25.85.141

Each number in the above array is an eight-bit integer (That is it is between 0 and 255 and called an Octet). There are three classes of networks in the Internet depending on the numbers of hosts it can handle. They are

Class A networks

Class B networks

Class C networks

Class A networks can handle a large number of hosts. Class B networks are capable of handling a moderate number of hosts.

Class C networks can handle only a small number of hosts. The maximum number of hosts handled by each network are tabulated in Table 1.4.1.

Table 1.4.1 Network Classes

Class of Network	*Maximum numbers of hosts a network can handle*
Class A	16777214 (256 * 256 * 256 = 16777216)
Class B	65534
Class C	254

1.4.1 Class A Networks

The first octet of a Class A network IP address represents the network ID and has a value from 1 to 126. So, there are only 126 Class A networks. However for each Class A network the second, third and the fourth octet represent the host ID, which can be any number between 0 and 256 and hence can have 16,177,214 hosts. (Note: 256 * 256 * 256 =16,177,216).

The general form of an IP address of a Class A network host is

N.H1.H2.H3.

Where

N ranges from 1 to 126

H1, H2, H3 are numbers from 0 to 255.

1.4.2 Class B Networks

The first octet of a Class B network IP address is a value from 128 to 191. The general format of a Class B network IP address is

N1.N2.H1.H2

Where

N1 ranges from 128 to 191

N2 ranges from 0 to 255

H1 and H2 may range from 0 to 255.

N1 and N2 form the network ID and H1 and H2 represent the host ID. There can be up to 16,384 Class B networks on the Internet, each of which can have up to 65,534 hosts.

(**Note:** N1 takes one of the 64 values from 128 to 191 and N2 takes 0 to 255. 254 * 254 = 65536)

1.4.3 Class C Networks

Class C networks are designed to handle the situation where there would be very large number of networks that contained a small number of networks which in turn contained a small number

of hosts. Small organizations having upto 254 hosts can go in for Class C networks. The general format of the IP address is

N1.N2.N3.H

Where

N1 ranges from 192 to 233

N2, N3 may vary from 0 to 255

H may vary from 2 to 255.

There can be upto 2,097,092 Class C networks on the Internet, each of which can have upto 254 hosts. The following table 1.3.2 summarizes the facts on the IP address formats of various Classes of networks on the Internet.

Table 1.4.2 Facts about the IP Addresses on the Internet

Class of the Network	*IP Address Format*	*First Octet Value*	*Maximum Numbers of Networks on the Internet*	*Maximum Numbers of Hosts on each Networks*
A	N.H1.H2.H3	1 to 126	126	16,777,214
B	N1.N2.H1.H2	128 to 191	16,384	65,534
C	N1.N2.N3.H	192 to 233	2,097,092	254

1.5 Domain Name

A name that identifies one or more *IP addresses*. For example, the domain name *microsoft.com* represents about a dozen IP addresses. Domain names are used in URLs to identify particular Web pages. For example, Consider the URL

`http://www.stxaviers.com/index.html`

The domain name is `stxaviers.com`. Every domain name has a suffix that indicates which *top level domain* (TLD) it belongs to. There are only a limited number of such domains. For example:

- **gov**—Government agencies
- **edu**—Educational institutions
- **org**—Organizations (nonprofit)
- **mil**—Military
- **com**—commercial business
- **net**—Network organizations

- **ca**—Canada
- **th**—Thailand

Because the Internet is based on IP addresses, not domain names, every Web server requires a *Domain Name System (DNS)* server to translate domain names into IP addresses as shown in Figure 1.7.

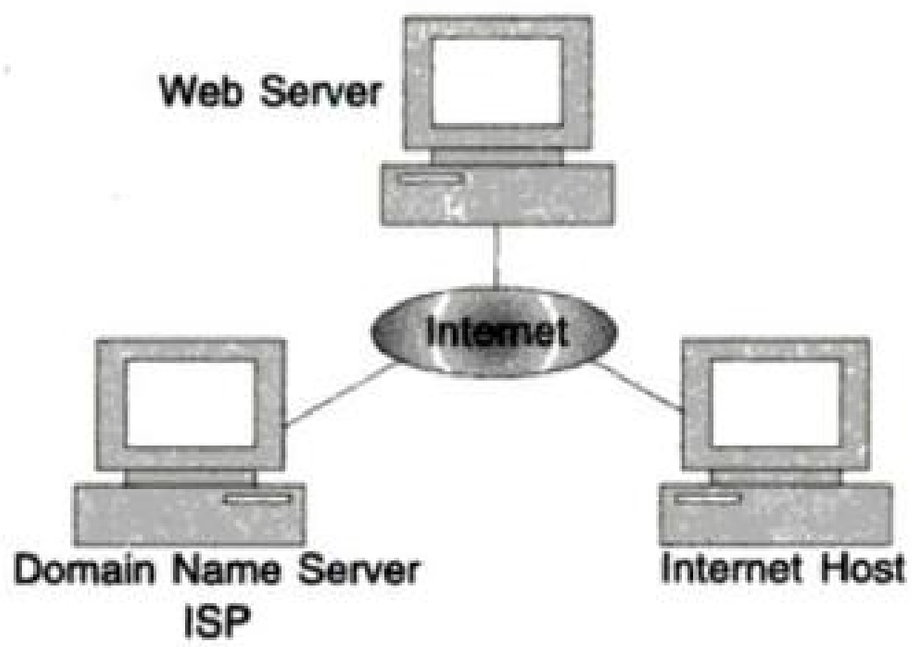


Figure 1.7 The Role of DNS

1.5.1 Uniform Resource Locator (URL)

URL is the abbreviation of *Uniform Resource Locator,* the global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located.

For example, the two URLs below point to two different files at the domain `stxaviers.com`. The first specifies an executable file that should be fetched using the FTP protocol; the second specifies a Web page that should be fetched using the HTTP protocol:

```
ftp://www.stxaviers.com/nov2002result.exe
http://www.stxaviers.com/index.html
```

1.6 Internet Services

In this section we discuss some important services available through Internet.

1.6.1 Gopher Service

The Internet *Gopher* protocol is designed primarily to act as a distributed document delivery system. While documents (and services) reside on many servers, Gopher client software presents

users with a hierarchy of items and directories much like a file system. In fact, the Gopher interface is designed to resemble a file system since a file system is a good model for locating documents and services

In essence, the Gopher protocol consists of a client connecting to a server and sending the server a selector (a line of text, which may be empty) via a TCP connection. The server responds with a block of text terminated with a period on a line by itself, and closes the connection. The server between transactions retains no state with a client. The simple nature of the protocol stems from the need to implement servers and clients for the slow protocols.

1.6.2 Instant Messaging

It is a type of communication service that enables you to create a private chat room with another individual. Typically, the instant messaging system alerts you whenever somebody on your private list is online. You can then initiate a chat session with that particular individual.

There are several competing *instant messaging systems*. Unfortunately, there's no standard.

1.6.3 Internet Relay Chat

A virtual room where a chat session takes place is called a chat room. If two computers are connected through Internet the *Internet Relay Chat* (IRC) application provides a platform for the users to chat. Technically, a chat room is really a channel, but the term *room* is used to promote the chat metaphor

1.6.4 Bulletin Board System (BBS)

BBS is an electronic message center. Most bulletin boards serve specific interest groups. They allow you to dial in with a modem, review messages left by others, and leave your own message if you want. Bulletin boards are a particularly good place to find free or inexpensive software products.

1.6.5 Usenet News Group Service

A worldwide bulletin board system that can be accessed through the Internet or through many online services is called a Usenet service. The Usenet contains more than 14,000 forums, called *newsgroups*, that cover every imaginable interest group. It is used daily by millions of people around the world.

1.7 Electronic Mail

Electronic mail is the transmission of messages over communications networks. The messages can be notes entered from the keyboard or electronic files stored on disk. Most mainframes, minicomputers, and computer networks have an e-mail system. Some electronic-mail systems are confined to a single computer system or network, but others have gateways to other computer systems, enabling users to send electronic mail anywhere in the world. Companies that are fully computerized make extensive use of e-mail because it is fast, flexible, and reliable.

Most e-mail systems include a rudimentary text editor for composing messages, but many allow you to edit your messages using any editor you want. You then send the message to the recipient by specifying the recipient's address. You can also send the same message to several users at once. This is called *broadcasting*.

Sent messages are stored in electronic *mailboxes* until the recipient fetches them. To see if you have any mail, you may have to check your electronic mailbox periodically, although many systems alert you when mail is received. After reading your mail, you can store it in a text file, forward it to other users, or delete it. Copies of memos can be printed out on a printer if you want a paper copy.

All online services and *Internet Service Providers (ISPs)* offer e-mail and also support gateways so that you can exchange mail with users of other systems. Usually, it takes only a few seconds or minutes for mail to arrive at its destination. This is a particularly effective way to communicate with a group because you can broadcast a message or document to everyone in the group at once.

Although different *e-mail systems* use different formats, there are some emerging standards that are making it possible for users on all systems to exchange messages. In the PC world, an important e-mail standard is MAPI. MAPI is the abbreviation of *Messaging Application Programming Interface*, a system built into Microsoft Windows that enables different e-mail applications to work together to distribute mail. As long as both applications are MAPI-enabled, they can share mail messages with each other. The recent addition to Email is the *Voice Email* in which a *Voice file* is communicated.

1.8 World Wide Web

Many people use the terms *Internet* and *World Wide Web* interchangeably, but in fact the two terms are not synonymous. The Internet and the Web are two separate but related things. The *Internet* is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. Information that travels over the Internet does so via a variety of languages known as protocols.

1.8.1 Web Technologies

The Web uses the HTTP protocol, only one of the languages spoken over the Internet, to transmit data. Web services, which use HTTP to allow applications to communicate in order to exchange business logic, use the Web to share information. The Web also utilizes browsers, such as Internet Explorer or Netscape, to access Web documents called Web pages that are linked to each other via hyperlinks. Web documents also contain graphics, sounds, text and video.

The Web is just one of the ways that information can be disseminated over the Internet. The Internet, not the Web, is also used for e-mail, which relies on SMTP, Usenet news groups, instant messaging and FTP. So the Web is just a portion of the Internet, albeit a large portion, but the two terms are not synonymous and should not be confused.

Web is a system of Internet servers that support specially formatted documents. The documents are formatted in a script called HTML (*HyperText Markup Language*) that supports links to other documents, as well as graphics, audio, and video files. This means you can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web. There are several applications called Web browsers that make it easy to access the World Wide Web. Two of the most popular being Netscape Navigator and Microsoft's Internet Explorer. In a web page information are presented using the following technologies:

- HTML
- Javascript
- Java Applets
- Java Server Pages (JSP)
- Active Server Pages (ASP)
- Java Servlets

In the above list the browser machine executes the first three whereas the last three run in the server machine. XML is a new technology, which is widely used in data formatting and communication. The web pages also contain objects such as cookies and plug-ins, which are explained in the following sections.

1.8.2 Plug-in

A hardware or software module that adds a specific feature or service to a larger system is called a plug-in. For example, there are number of plug-ins for the Netscape Navigator browser that enable it to display different types of audio or video messages.

1.8.3 Cookie

Cookie is a message given to a Web browser by a Web server. The browser stores the message in a text file (in the client machine itself) and uses each time the browser requests a page from the server. The main purpose of cookies is to identify users and possibly prepare customized Web pages for them. When you enter a Web site using cookies, you may be asked to fill out a form providing such information as your name and interests. This information is packaged into a cookie and sent to your Web browser. The next time you go to the same Web site, your browser will send the cookie to the Web server. The server can use this information to present you with custom Web pages. So, for example, instead of seeing just a generic welcome page you might see a welcome page with your name on it. The name *cookie* derives from UNIX objects called *magic cookies*. These are tokens that are attached to a user or program and change depending on the areas entered by the user or program.

Cookies are messages that a Web server transmits to a Web browser so that the Web server can keep track of the user's activity on a specific Web site. The message that the Web server conveys to the browser is in the form of an HTTP header that consists of a text-only string. The text is entered into the memory of the browser. The browser in turn stores the cookie information on the hard drive so when the browser is closed and reopened at a later date the cookie information is still available.

Web sites use cookies for several different reasons:

- To collect demographic information about who is visiting the Web site. Sites often use this information to track how often visitors come to the site and how long they remain on the site.
- To personalize the user's experience on the Web site. Cookies can help store personal information about you so that when you return to the site you have a more personalized experience. When you visit a web site again and see your name mysteriously appear on the screen, it is because, on a previous visit you gave your name to the site and it was stored in a cookie. When you returned you would be greeted with a personal message. A good example of this is the way some online shopping sites will make recommendations to you based on previous purchases. The server keeps track of what you purchase and what items you search for and stores that information in cookies.
- Web sites will often use cookies to keep track of what ads it lets you see and how often you see ads.

Cookies do not act maliciously on computer systems. They are merely text files that can be deleted at any time. They are not plug-ins nor are they programs. Cookies cannot be used to spread viruses and they cannot access your hard drive. This does not mean that cookies are not relevant to a user's privacy and anonymity on the Internet. Cookies cannot read your hard drive to find out information about you; however, any personal information that you give to a Web site, including credit card information, will most likely be stored in a cookie unless you have

turned off the cookie feature in your browser. The cookie will only contain information that you freely provide to a Web site.

Cookies have six parameters that can be passed to them:

- The **name** of the cookie.
- The **value** of the cookie.
- The **expiration date** of the cookie—this determines how long the cookie will remain active in your browser.
- The **path** the cookie is valid for—this sets the URL path the cookie is valid in. Web pages outside of that path cannot use the cookie.
- The **domain** the cookie is valid for—this takes the path parameter one step further. This makes the cookie accessible to pages on any of the servers when a site uses multiple servers in a domain.
- The need for a **secure** connection—this indicates that the cookie can only be used under a secure server condition, such as a site using SSL.

Both Netscape and Microsoft Internet Explorer (IE) can be set to reject cookies if the user prefers.

1.9 Internet Security

As the Internet connects millions of computers across the globe, the security and privacy are two vital issues in this new era of Information Technology. This section introduces two widely-used security technologies of the web, SSL and S-HTTP.

1.9.1 Secure Sockets Layer (SSL) and Secure HTTP (S-HTTP)

Secure Sockets Layer, is a protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a public key to encrypt data that's transferred over the SSL connection. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with `https:` instead of `http:`.

Another protocol for transmitting data securely over the World Wide Web is *Secure HTTP (S-HTTP)*. Whereas SSL creates a secure connection between a client and a server, over which any amount of data can be sent securely, S-HTTP is designed to transmit individual messages securely. SSL and S-HTTP, therefore, can be seen as complementary rather than competing technologies. The Internet Engineering Task Force (IETF) has approved both protocols as a standard.

1.9.2 Encryption

The word *encryption* means translation of data into a secret code. Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to *decrypt* it. Unencrypted data is called *plain text* ; encrypted data is referred to as *cipher text*.

There are two main types of encryption. They are

- Asymmetric encryption (also called public-key encryption) and
- Symmetric encryption.

1.9.3 Asymmetric Encryption (public-key encryption)

Whitfield Diffie and Martin Hellman invented public key cryptography in 1976. For this reason, it is sometime called *Diffie-Hellman encryption.*

This is a cryptographic system that uses two keys—a *public key* known to everyone and a *private* or *secret key* known only to the recipient of the message. When A wants to send a secure message to B, he uses B's public key to encrypt the message. B then uses his private key to decrypt it. An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.

Public-key systems, such as *Pretty Good Privacy (PGP)*, are becoming popular for transmitting information via the Internet. They are extremely secure and relatively simple to use. The only difficulty with public-key systems is that you need to know the recipient's public key to encrypt a message for him or her. What's needed, therefore, is a global registry of public keys, which is one of the promises of the new LDAP technology.

1.9.4 Symmetric Encryption

In Symmetric encryption the same key is used to encrypt and decrypt the message.

1.10 Electronic Commerce (E-Commerce) and Electronic Data Interchange (EDI)

EDI stands for the transfer of data between different companies using networks, such as the Internet. As more and more companies get connected to the Internet, EDI is becoming increasingly important as an easy mechanism for companies to buy, sell, and trade information. ANSI

has approved a set of EDI standards known as the *X12 standards*. E-Commerce refers to transacting business on-line. This includes, for example, buying and selling products with digital cash and via Electronic Data Interchange (EDI).

1.10.1 Digital Cash

Digital cash refers to a system that allows a person to pay for goods or services by transmitting a number from one computer to another. Like the serial numbers on real dollar bills, and rupee bills the digital cash numbers are unique. Each one is issued by a bank and represents a specified sum of real money. One of the key features of digital cash is that, like real cash, it is anonymous and reusable. That is, when a digital cash amount is sent from a buyer to a vendor, there is no way to obtain information about the buyer. This is one of the key differences between digital cash and credit card systems. Another key difference is that a digital cash certificate can be reused.

Digital cash transactions are expected to become popular shortly. However, there are a number of competing protocols and it is unclear which ones will become dominant. Most digital cash systems start with a participating bank that issues cash numbers or other unique identifiers that carry a given value, such as five dollars. To obtain such a certificate, you must have an account at the bank; when you purchase digital cash certificates, the money is withdrawn from your account. You transfer the certificate to the vendor to pay for a product or service, and the vendor deposits the cash number in any participating bank or retransmits it to another vendor. For large purchases, the vendor can check the validity of a cash number by contacting the issuing bank

1.10.2 Digital Signature

A digital code that can be attached to an electronically transmitted message that uniquely identifies the sender is called a digital signature.. Like a written signature, the purpose of a digital signature is to guarantee that the individual sending the message really is who he or she claims to be. Digital signatures are especially important for electronic commerce and are a key component of most authentication schemes. To be effective, digital signatures must be unforgeable. There are a number of different encryption techniques to guarantee this level of security.

Exercises

1.1 Write a note on *SMTP*

1.2 In Internet, use yahoo.com to search for "Internet Protocols" and find the result

1.3 List some popular Search Engines in the Internet. Discuss with your friends on the Search Engines in Internet

1.4 Who are the prominent Internet Service Providers in your country

1.5 In Internet what is dial-up connectivity

1.6 What is broad band Internet connectivity. Know more on this topic from your friends or from Internet

1.7 What is DSL Internet Connectivity.

1.8 What is *Optical Fibre* network.

1.9 What are the technologies used in web design.

1.10 What is the difference between *web design* and *web enabling*.

1.11 Write a note on *Internet Security*

1.12 Explain the following:

a. E-mail
b. E-Commerce
c. E-Cash
d. E-Banking
e. E-Police
f. Mobile Computing
g. WAP
h. Short Message Service
i. IP Services
j. VoIP

1.13 Write a note on *Electronic Signature*.

1.14 Write a note on *Cyber Law*.

CHAPTER 2

Introduction to HTML

HTML is the basic tool for designing a web page. HTML is the acronym for *Hypertext Mark up Language*. It is a documentation language to mark the headings, title, table and forms.

2.1 A Brief History

IBM wanted to set a documentation system in which we can mark the title, headings, paragraphs and font type selections in the 1980s. They came out with a set of mark-up system called it *General Mark Up Language (GML)*. In 1986, International Standardizing Organization(ISO) took up this concept and standardized it as *Standard Generalized Mark Up Language (SGML)*. In 1989 Tim Berners Lee and his team in the European Laboratory for Particle Physics (CERR) designed the present form of the documentation language and called it HTML.

2.2 HTML Tags

When a web page is to be developed, the following are to be planned:

- Content of the page
- Appearance of the page

The appearance of the web page is coded in HTML language using HTML tags.

2.2.1 HTML Tags and their Attributes

An HTML tag is a word specifying the appearance of the content. The following are the salient features of an HTML tag:

- A tag is enclosed between the < and > symbols. Examples are `<head>`, `<h1>`, `<ol>`.

- Most of the tags have end tag, which begins with </. For example </h1> is the end tag for `<h1>`.

A tag may have some attributes. Attributes are the properties of the tag. For example, consider the <hr> tag. This tag is for drawing the horizontal ruling (horizontal line). This tag contains the following attributes.

1. Size(thickness of the line)
2. Width
3. Alignment

These are represented as follows:

```
<hr size=10 width=25% align = LEFT>
```

2.3 HTML Documents

Every html document begins with the `<html>` tag and ends with the corresponding end tag `</html>`. A document has the following two sections.

1. Head
2. Body

The head section begins with the `<head>` tag and ends with `</head>` tag. In the head section title is the most important item. The title begins with `<title>` and ends with `</title>`.

2.4 Header Section

Every HTML document must have a head section which begins with the tag `<head>` and ends with `</head>`. The following are some important components of the head section.

1. Title
2. Prologue
3. Links

2.4.1 Title Tag

The title is the heading that appears as the title of the window. It is enclosed between the tags `<title>` and `</title>`.

2.4.2 Prologue

Prologue is only a comment, which can tell about the HTML version that is adopted for preparing the document. A prologue is given as shown below:

```
< ! Doctype HTML 4.0 >
```

2.4.3 Links

The Link tag can be used for the following purposes:

- To inform the browser the previous document
- To inform the browser the next document
- To link the *banner*
- To inform the location of the base document location

Previous and Next Attributes

The HTML document can be placed in between two HTML document using LINK tag. Suppose we want to assign a HTML document `nov2002.html` as the previous document and `jan2003.html` as the next document. We define as follows:

```
<head>
<title> travel in december 2002 </title>
<link rel = previous href ="nov2002.html">
<link rel = next href="jan2003.html">
</head>
```

Consider the following:

```
<link rel = home href="webtechnologybook.html">
```

The above statement assigns `webtechnologybook.html` for the Home button of the tool bar.

Banner Attribute

A banner is a fixed part of the page that will stay on the screen when we scroll the text of the page. The emblem of the company, or the title of the company can be a banner. The banner must be prepared as a separate HTML document and it must be linked as banner to the document. For example, Suppose logo.html is a HTML document which shows the emblem of the college.

We can link it to a HTML document in the Header section as follows:

```
<head>
<title> travel in December 2002 </title>
<link rel = banner href="logo.html">
</head>
```

2.5 Body Section

The body begins with `<body>` and ends with `</body>`. The following is a simple template for creating a HTML document.

```
<html>
<head>
<title>
The title appears here
</title>
</head>
<body>
.......
.......
</body>
</Head>
</HTML>
```

The `<body>` tag has several attributes. The following is a partial list.

- Background design `(background)`
- Background color `(bgcolor)`
- Text color `(text)`
- Link color `(link)`
- Active link color `(alink)`
- Visited link color `(vlink)`

2.6 Headings

Headings can be created with tags h1, h2,...h6. H1 will make a bigger heading, h2 will make a smaller one and h3 will make still a smaller heading and so on. For example, if you want the name of your company (Innovative Educational Media Ltd.) to appear as a big heading, we type as follows

```
<h1>Innovative Educational Media Ltd</h1>.
```

The output will be as shown in Figure 2.1.

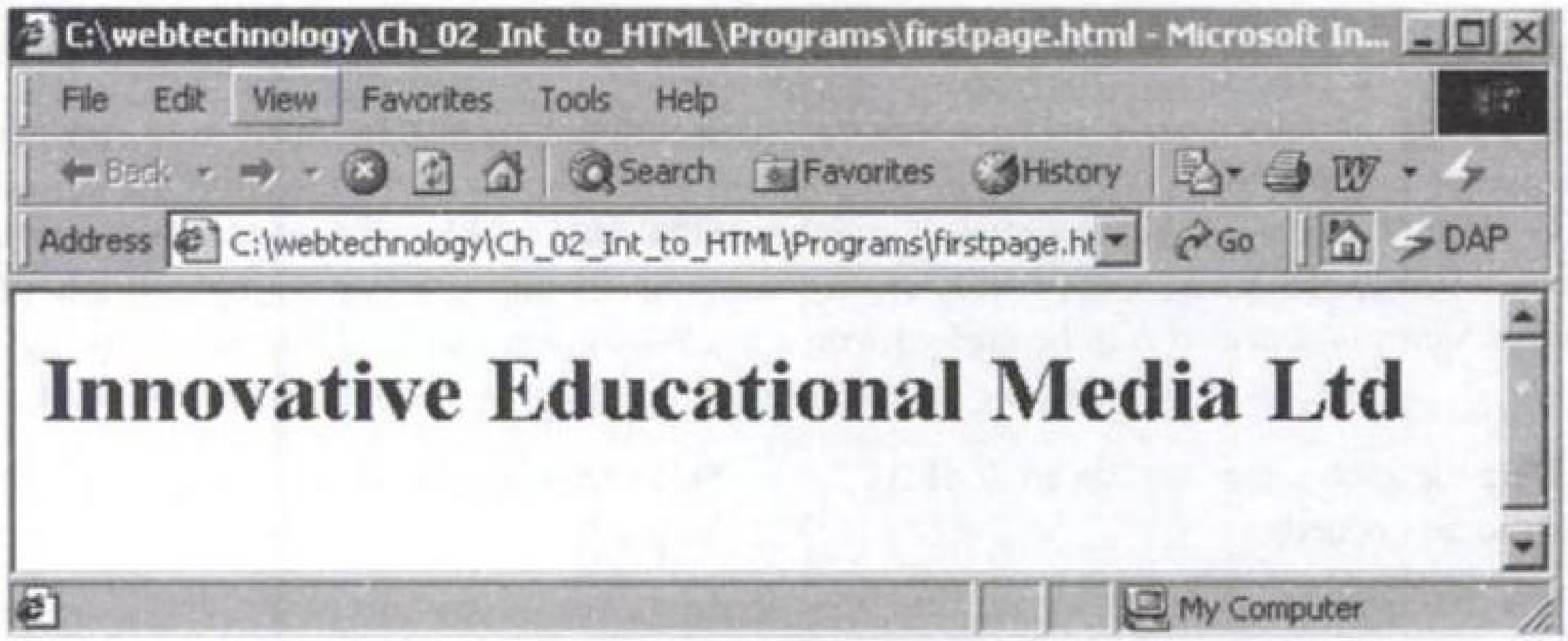


Figure 2.1 Heading Tag <h1>

Suppose a document has the following lines:

```
<h1> Innovative Educational Media <h1>
<h2> Innovative Educational Media <h2>
<h3> Innovative Educational Media <h3>
<h4> Innovative Educational Media <h4>
<h5> Innovative Educational Media <h5>
<h6> Innovative Educational Media <h6>
```

The output of the above document is shown in Figure 2.2.

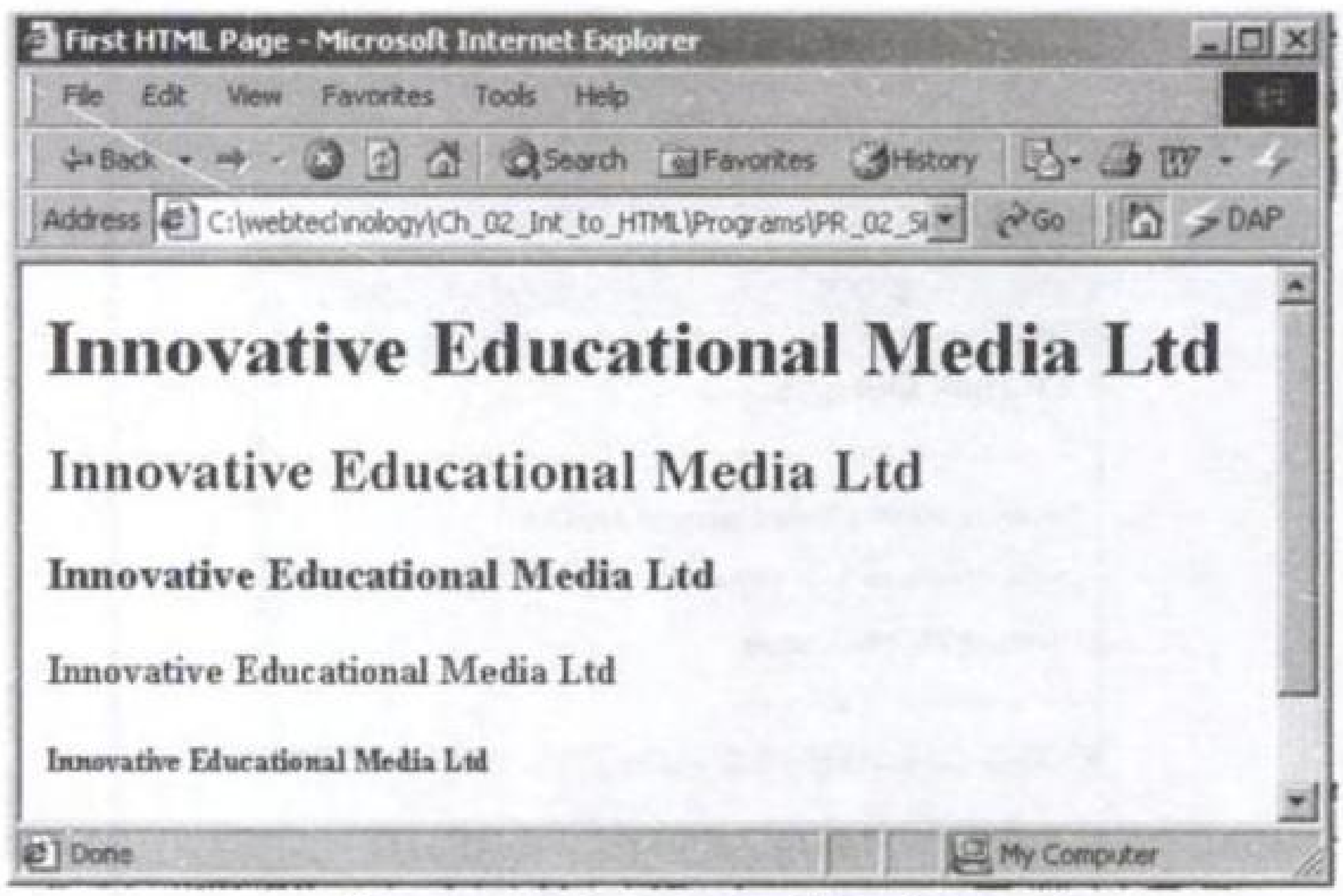


Figure 2.2 Six Levels of Headings

2.7 Link Documents using Anchor Tag

When a page is shown, some of its words may need further explanation. Such words are called *hot text* or *link* and they appear in a different color. When the cursor is moved to the *hot text* a hand symbol appears. When we click it, another HTML file will be opened and that will have the required explanation. A hot text is created with *anchor tag* `<a>`. For example, if a hot text *United States of America* is to be created with a html document `us.html`, we must type as

```
<a href="us.html"> United States of America </a>.
```

Here the anchor tag `<a>` has an attribute `href`. This attribute specifies the name of the document to be opened.

For example, a HTML file `travel.html` is shown below. It contains details about the travel made.

```
<html>
<head>
<title>Travel in December 2002 </title>
</head>
<h1>Travel Details </h1>
<hr>
<br>
<h3>December 28, 2002 <a href="us.html"> United States of America </a> </
h3>Quarterly Meeting with Bank of America
<br><h3> December 29, 2002 <a href= "canada.html" >Canada </a></h3> Meet-
ing with Brach office Managers<br>
</body>
</HTML>
```

When this file is opened using an Internet browser we get the page shown in Figure 2.3.

Figure 2.3 First Web Page

In this page *United States of America* is a hot text. When this is clicked, it will open `us.html`.

2.9 Font Tag

The <font> tag is used to set a specific font and size. It usually has two attributes, namely face and size.

For example, consider the following:

```
<font face=TimesRoman size = 35>
Web Technology book by Dr. C. Xavier
</font>
```

2.10 Images and Pictures

A picture or an image in the web page can be inserted using the <img> tag. The <img> tag has several attributes to inform the source, height of the picture, width of the picture alignment etc. The following are its important attributes:

- src
- Height
- Width
- Align
- alt

The src attribute specifies the source of the picture. It gives the file name of the picture file. The height and the width may be expressed in terms of either pixels or percentage or en units. The default is the pixels. For example consider the following:

```
<img src="companylogo.gif" height=100 width=120>
```

The align attribute an be used to align the picture.

2.11 Listing

This section introduces two more tags <ol> and <ul> that lists a set of items either with serial numbers or with bullets. They are called *Ordered* and *Unordered lists*.

2.11.1 Unordered Lists

The Unordered list is represented by `<ul>` tag and `</ul>` tag. `<ul>` is given at the beginning and

```
<tr>
<td>Dec 28,2002<td>USA <td>Quarterly Review Meeting
<tr>
<td>Dec 29,2002<td>Canada <td>Branch Managers Meeting
</table>
</body>
</html>
```

The output is shown in Figure 2.6.

Figure 2.6 Table Showing Places Visited

Exercises

2.1 Design a HTML page describing your Profile in one paragraph. Design in such a way that it has a heading, a horizontal rule, th`ree links and your photo. Also, write three HTML documents for the links.

2.2 Design a HTML page describing your academic career. The page will tell about the degrees, institutions and your hobbies. Add some lists also.

2.3 Design a HTML page on your native place?

2.4 Design a HTML page on your friends. List your friends. Each friend's name is a link. Prepare separate HTML documents on each friend and call them in the appropriate link?

2.5 Prepare a HTML page listing popular car companies. For each company prepare a sub-list showing various brands of cars it offers.

3.2 Frame Definition

The definition of the frame is given using the `<frame>` tag. The `<frame>` tag may have any of the following attributes:

- Source html address (src)
- Name of the frame (name)
- Margin width (marginwidth)
- Scrolling button (scrolling)
- Can it be resized (noresize)

Example 3.2.1 Banking Self Help Page

Consider the following frameset definition. Two rowwise frames are defined.

```
<html>
<head>
<title>
Welcome to Commercial Bank of Singapore
</title>
</head>
<frameset rows="45,55">
<frame name="f1" src="D:\servlets\Banking\Bank1.html">
<frame name="f2" src="http:\\localhost:8080\servlet\BankSelfHelpServlet">
</frameset>
</html>
```

The output appears as shown in Figure 3.1

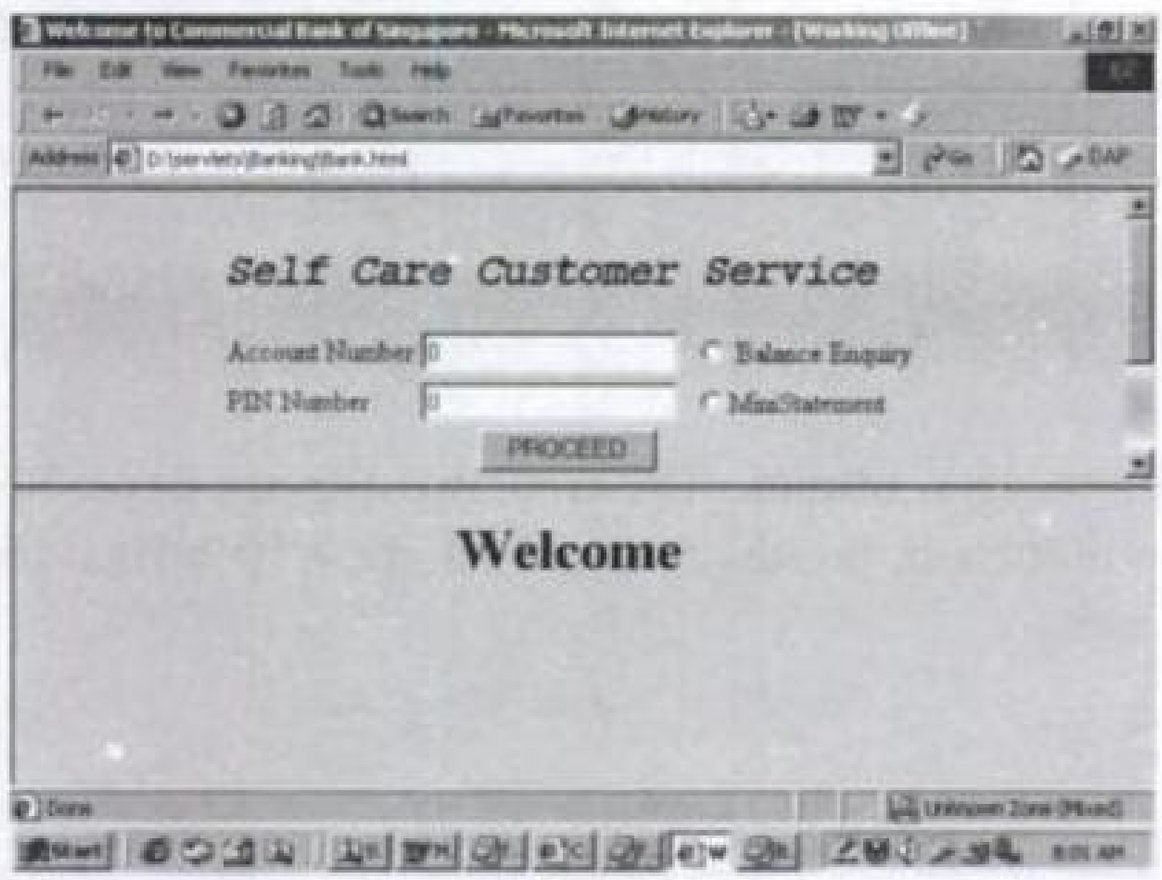


Figure 3.1 Banking Self Help Page

In the case of method = "get", the data received in the form are placed at the end of URL. If the form is very big and gets a number of input, the "get" method causes the URL to be very long. So, the method = "get" option is often discouraged.

3.4.3 ENCTYPE Attribute

This attribute is used to inform the server the way to handle the encryption process. Usually it is set to

application/x-www-form-urlencoded. That is, the value is given as

```
enctype = "application/x-www-form-urlencoded"
```

3.5 Elements of a Form

In a form there can be several elements to get the input form the use. They are

1. Selection list box
2. Input box
3. Text Area

3.5.1 Selection List Box

A selection list presents a list of options to the user. The user can select his choice from the list. The selection list box is created with <select> tag. The definition ends with </select> tag.

The <select> tag has three attributes. They are

1. Name attribute
2. Size attribute
3. Multiple attribute

The name attribute assigns a name for the variable, which will hold the selected choice. For example consider the following:

```
<select name="nameBox">
<option>Aparna</option>
<option>Nithya</option>
<option>Priya</option>
</select>
```

3.5.5 Password Field

The following HTML document shows a password field. Notice that when a user types the password, the characters are no displayed. The asterisk symbol will be displayed.

```
<html>
<head>
<title>Authorisation Page</title>
</head>
<body>
<form name="myform">
User Id :<input type=text name=id><br>
Password :<input type=password name=pword><br>
<input type=submit value=Go>
  </form>
</center>
</body>
</html>
```

The output is shown in Figure 3.6.

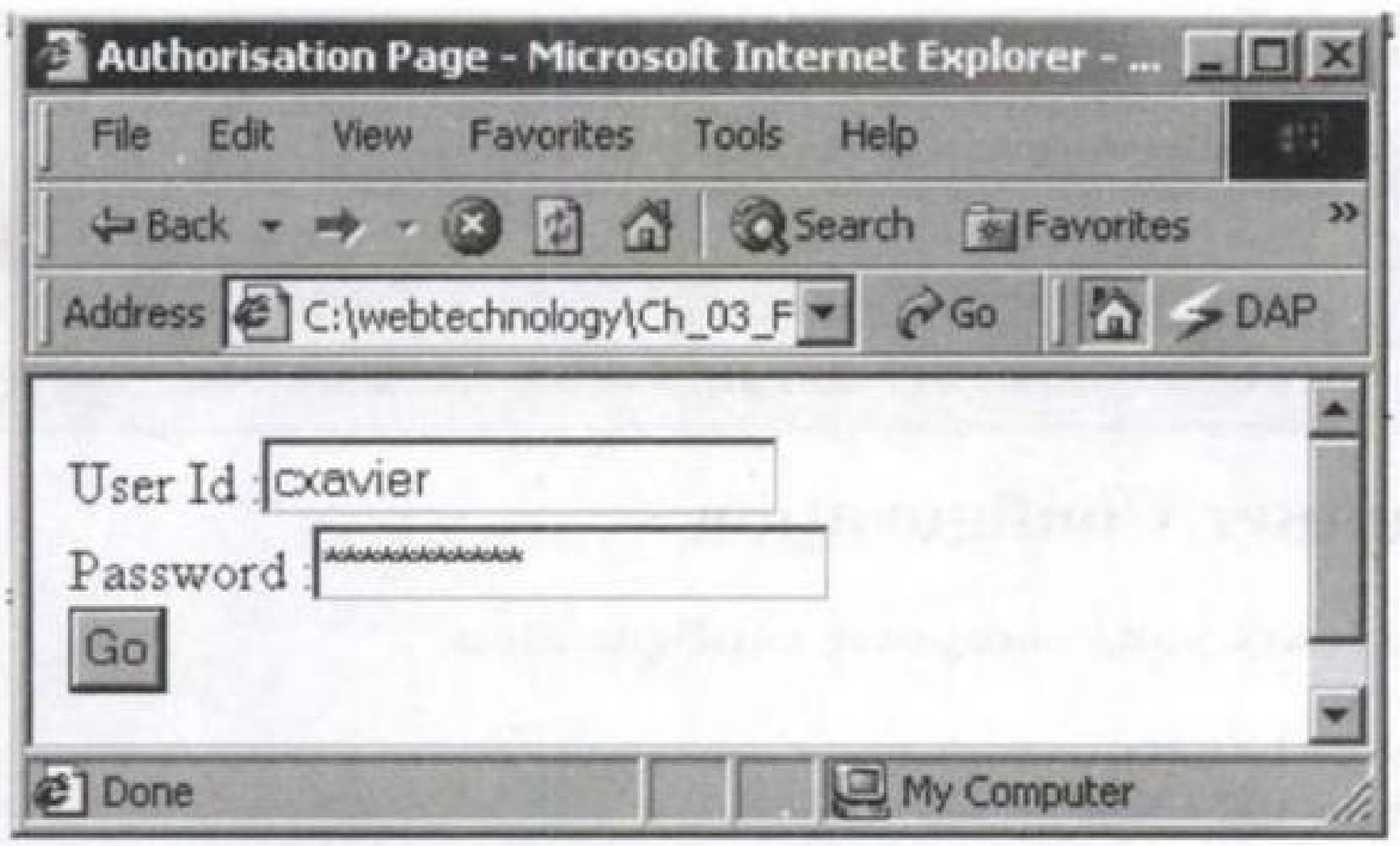


Figure 3.6 Password Field

Example 3.5.1 Changing the Password

This example illustrates the form for changing the password. This shows three password fields, one for old password and two for the new password. The HTML document is shown below:

```
<html>
<head>
<title>Password Change Page-1</title>
```

Example 3.5.3 Banking Web Page

This example illustrates how a banking self care web page be designed. The user must type the Account number and the PIN number(Personal Identification Number), The account number can be typed in a text field. The PIN number is a secret code and hence has to be typed in a password field. The various services available to the customer are shown as radio buttons.

```
<html>
<head>
</head>
<body bgcolor=bisque text=black>
<form action=http:\\localhost:8080\servlet\BankSelfHelpServlet target=f2>
<i><b><h2>Self Care Customer Service</h2></b></i>
<table align=center>
<tr>
<td>Account Number</td>
<td><input type=text name=accno ></td>
<td><td><input type=radio name=service>Balance Enquiry</td>
</tr>
<tr>
<td>PIN Number </td>
<td><input type=password name=pin value=0 ></td>
<td><td><input type=radio name=service>MiniStatement</td>
</tr>
</table>
<input type=button value=PROCEED >
<input type=hidden name=serviceRequest value=0>
</form>
</body>
</html>
```

The output is shown in Figure 3.10.

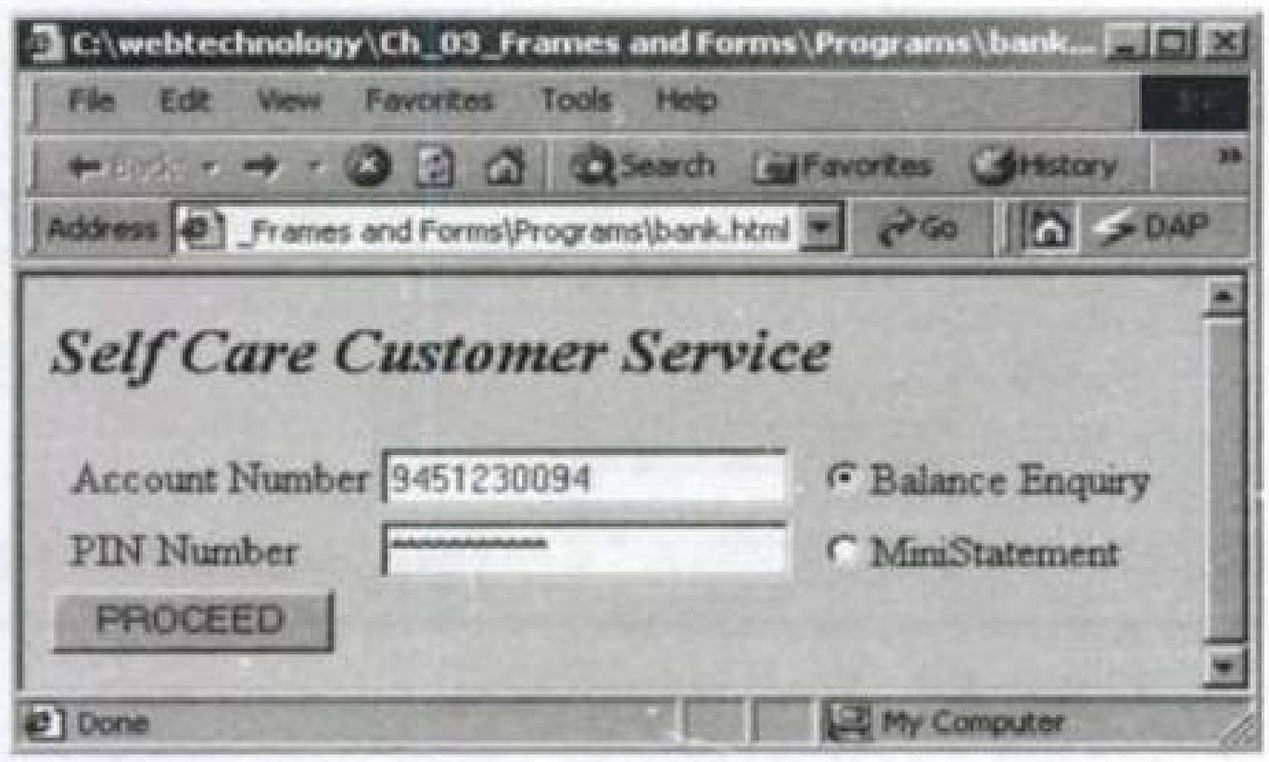


Figure 3.10 Banking Web Page

CHAPTER 4

Elements of JavaScript

Though JavaScript is a free flow language it is vital to know the basic framework of the language. These fundamental elements of the language are similar to that of any other language. But for the sake of completeness, this Chapter deals with variables, data types, operators, expressions and statements.

4.1 Data Types

Due to the use of JavaScript in interactive web design, the main focus of the language is not processing of numeric data for precision. However the language supports *Numeric*, *Boolean* and *String* data. Apart from the above three types, Object is another important data type supported. As a open standard the *Null Data* and *Undefined* data types are also supported.

4.1.1 Numbers

The integer and floating point numeric data are called *numbers*. A number can be positive, negative or zero. Integers can be represented in decimal, octal and hexadecimal form. The octal integer begins with a zero and hexadecimal integer begins with Ox or oX. The following are some valid integers.

45	Decimal value 45
1	Decimal value 0
–25	Decimal value –25
–015	Octal value $(-15)_8$
OX2A	Hexadecimal value $(2A)_{16}$

After executing the statement

$x + = 2y + z + 100$

the values will become, $x = 180$, $y = 20$, $z = 30$. So += is also considered as an assignment operator. Table 4.1 gives a list of assignment operators:

Table 4.1 Assignment Operators

Assignment Operator	*Name and Example*	*Meaning*
+ =	Addition assignment	
	`x += 5`	`x = x + 5`
− =	Subtraction Assignment	
	`x - = 5`	`x = x - 5`
*=	Multiplication Assignment	
	`x *= 5`	`x = x * 5`
/=	Division Assignment	
	`x /= 5`	`x = x / 5`
%=	Modulo Assignment	
	`x %= 5`	`x = x % 5`

4.3.6 Conditional Assignment Operator

The conditional operator checks a condition and then assigns a value to a variable. The general syntax of the conditional assignment operator is shown below:

```
Variable = (condition) ? Value 1 : Value 2;
```

Where condition is a logical expression

`Value 1` is the value assigned to the variable when the logical expression is true.

`Value 2` is the value assigned to the variable when the logical expression is false.

For example consider the following:

```
String result;
result = (mark >= 40) ? "Pass" : "Fail";
```

In this example the value "`Pass`" is assigned when mark is greater than or equal to 40. Otherwise "`Fail`" is assigned.

```
            resa=hex(a);
            resb=hex(b);
            resc=hex(c);
            d=resa+resb+resc
            document.bgColor="#"+d;
            }
        else
         alert("Enter values between 0 & 255 for red,blue,green");
        }
        else
         alert("Enter 3-digit number for red,blue,green");
        }
        function hex(a1)
         {
         b1=Math.floor(a1/16);
         c1=a1%16;
         switch(b1)
         {
          case 10:
           b1='A';
           break;
          case 11:
           b1='B';
           break;
          case 12:
           b1='C';
           break;
          case 13:
           b1='D';
           break;
          case 14:
           b1='E';
           break;
          case 15:
           b1='F';
           break;
          default:
           break;
          }
          switch(c1)
         {
          case 10:
           c1='A';
```

4.6 Date Objects

JavaScript does not have a date datatype. Date Objects and its methods are used to work with dates. The Date Object has methods for setting, getting and manipulating dates. JavaScript handles dates similar to Java. To create a date object:

```
DateObject name = new Date ([parameters]);
```

Passing arguments are optional. The parameters can be any of

- A string representing a date such as
  ("February 15, 2000 09:08:15). The format is "Month day, year, hours : minutes : seconds."
- An integer of values for writing year, month and day or a set of values for year, month, day, hours, minutes and seconds.

The methods used with Date Objects are:

Set — used to set date and time values in Date Objects.

get — getting date and time values from Date Objects.

to — Returning string values from Date Objects.

Assume the following example:

```
mydate = new Date ("February 15, 2000")
```

Then mydate `getMonth()` will contain 1 and `getDate()` contains 15.

The detailed list is given below in Table

Methods	*Description*

`get Date()`
`get UTC Date()`
`set Date()`
`set UTC Date()`
	Returns or sets the day of month of the Date Object

`get Day()` `get UTC Day()`	Returns the day of the week of the Date Object

`get Hours()`
`get UTC Hours()`
`set Hours()`
`set UTC Hours()`
	Returns or sets the hour of the Date Object

`get Milliseconds()`
`get UTC Milliseconds()`
`set Milliseconds()`
`set UTC Milliseconds()`
	Returns or sets the milliseconds value of the Date Object

(Contd.)

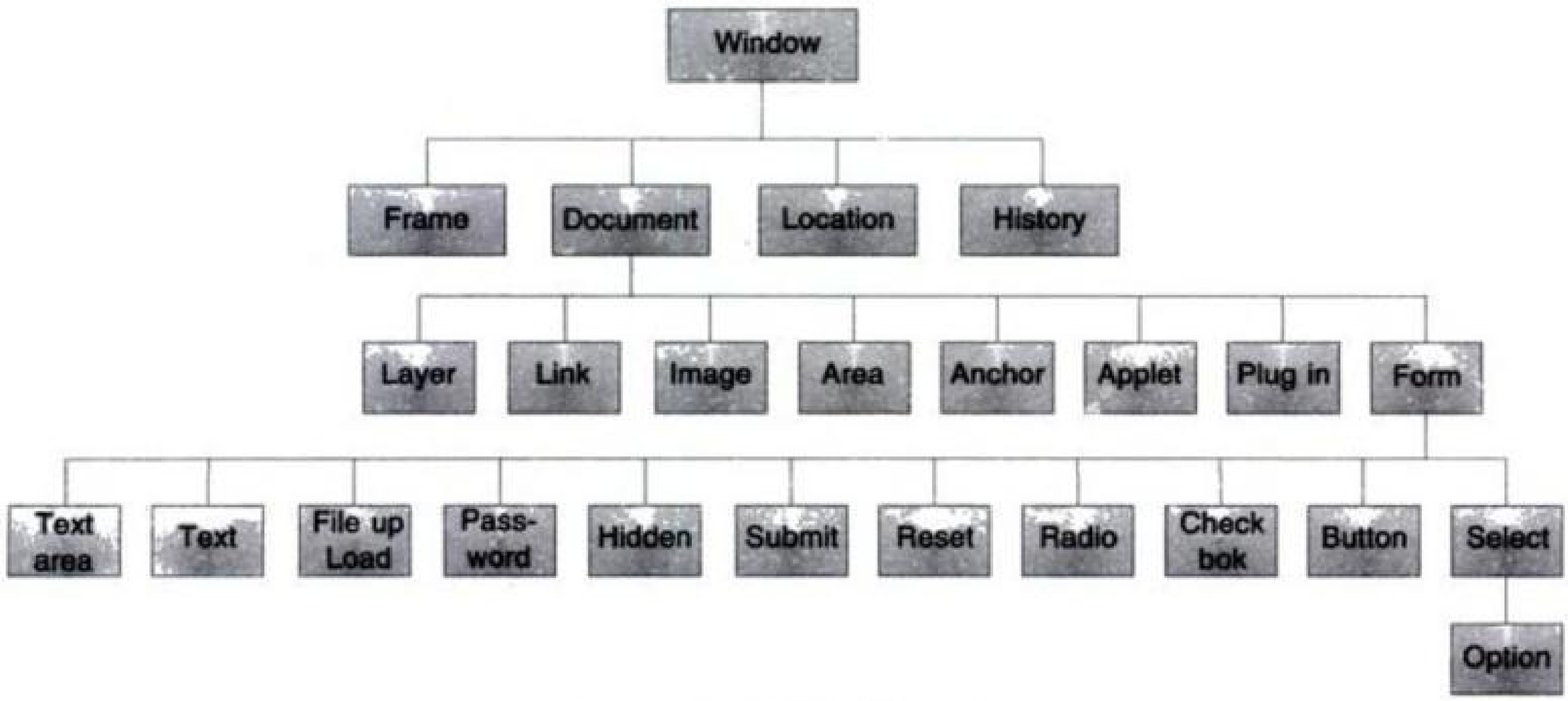


Figure 5.2 DOM Hierarchy

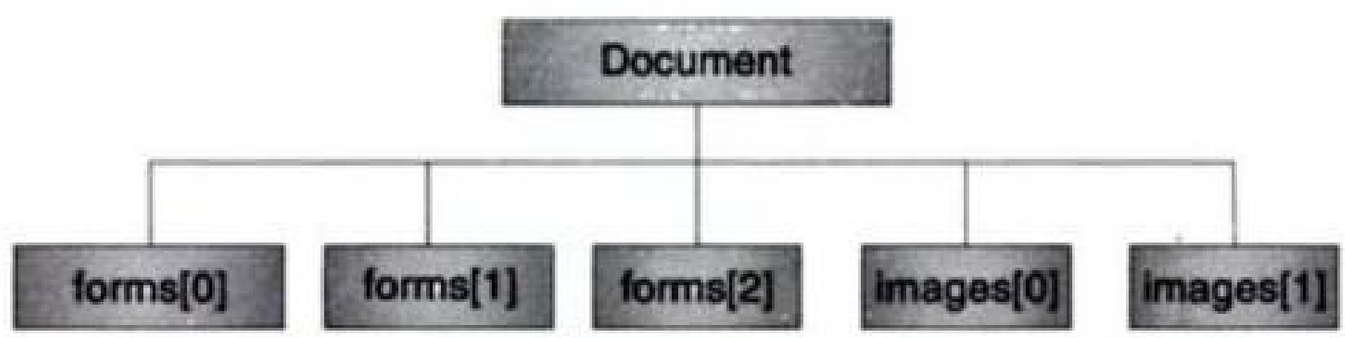


Figure 5.3 The Reference of the Document Objects

In the next section let us study how different objects in a document are referred.

5.2 The Document Object

The document object refers to the current HTML/Javascript document displayed by the browser. It has several properties including the following:

```
document.bgcolor
document.background
document.font
```

It is possible to dynamically change the properties of the document. For example, consider the following function:

```
function changecolor(c)
{
document.bgcolor = c
}
```

5.4 Forms and Elements

This section introduces another important object of a document, Forms. Forms play a major role in the interaction between the user and the browser. The user types the input in forms. A HTML document may consist of several forms. Each form may consist of several Elements, input objects. For example the following are some elements.

- Button
- Submit button
- Radio button
- Check box
- Select list box
- Text input field
- Text Area

In a form these elements are referred to as an array called `elements[]`. For example consider the form shown in Figure 5.7.

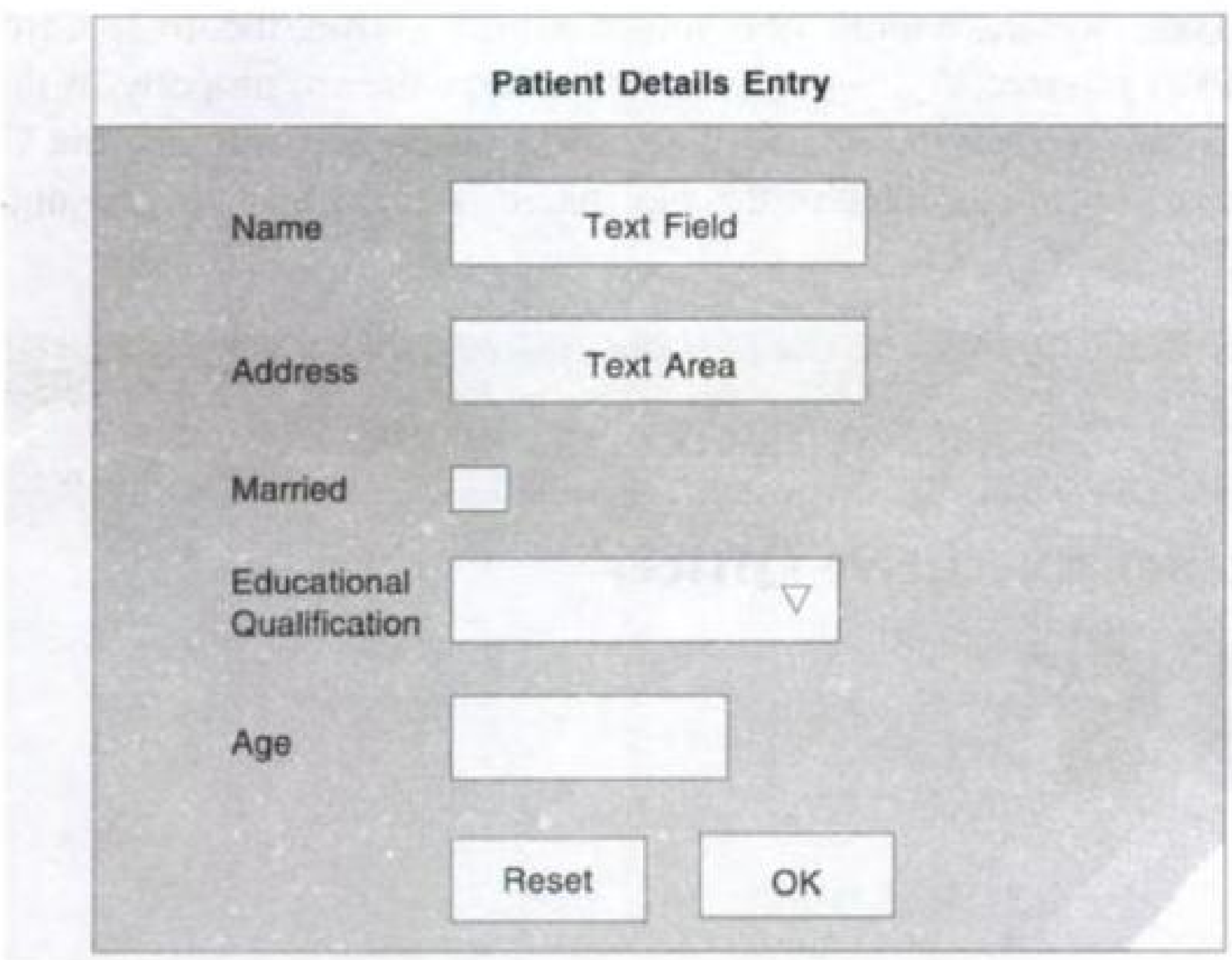


Figure 5.7 Patient Details Entry

This form has seven elements. JavaScript represents them using an array `elements[]`. Table 5.4.1 shows the reference details of each of the elements.

We can call this method when the OK button of Television order form is clicked.

```
<input type = button onClick = "alertTV() " >
```

Similarly functions `alertRefrigerator()` and `alertWashingmachine()` are also shown in the following document .

```
<html>
<head>
<title> Order form </title>
<script language="JAVASCRIPT">
<!-
function alertTV()
{
  size=document.tvform.tvsize.value
alert("You want to order a television of size = "+size)
}
function alertRef()
{
  size=document.refform.refcapacity.value
  alert("You want to order a Refrigerator of capacity = " +size )
}
function alertWashing()
{
  size=document.washform.washcapacity.value
alert("You want to order a Washing Machine of capacity = "+size)
}
//->
</script>
</head>
<body>
<form name=tvform >
<h2> Television order </h2>
   Type the screen size
<input type = text name=tvsize >
<input type = button value = " OK " onClick="alertTV()";>
</form>
<hr>
<br>
<form name=washform >
<h2> Washing machine order </h2>
   Type the capacity <input type = text name="washcapacity">
<input type = button value = " OK " onClick="alertWashing()";>
</form>
```

The HTML document with the Javascript function is shown below:

```
<html>
<head>
<title>Telephone Directory</title>
</head>
<body>
<script>
var i,f,phone;
function getnumber(f)
{
phone=new Array(6184723,7284131,4214852,4718912,5151082,
6272852,2343842,6262719,8432233,8187235)
i=f.nameBox.selectedIndex;
f.PhoneNumber.value=phone[i]
}
</script>
<form name=telephone form>
<pre><center><hr>
<h2> TELEPHONE DIRECTORY </h2>
<pre><center><hr><br>
select the name and press the button below to get the telephone number
  <br>
NAME :<select name="nameBox">
<option>Aparna</option>
<option>Nithya</option>
<option>Priya</option>
<option>Sandhya</option>
<option>Shruthi</option>
<option>Sumathi</option>
<option>Gowri</option>
<option>Krithika</option>
<option>Rekha</option>
<option>Reena</option></select><br>
PHONE NUMBER :<input type=text name=PhoneNumber><br>
<input type=button value="GetNumber"
onClick=getnumber(this.form)>
<br><br><hr><br><br>
</form>
</body>
</html>
```

The output is shown in Figure 5.16.

In general an event is a relationship between the user and the web document. Javascript defines the set of events and a mechanism to handle that.

5.5.1 Event Object

Javascript considers every event as an object and hence with some properties. The properties of the event object contains the information (details) about the event.

Table 5.5.1 illustrates some events and the properties.

Table 5.5.1 Some Events and Their Properties

Event	*Property*
The user clicks the left button of the mouse.	(X, Y) co-ordinate of the point of click with reference to the screen.
The user presses a key.	ASCII value of the key pressed.

Table 5.5.2 shows a list of events recognized by Javascript. Notice that the events are specific to the browser .So, we use the symbol NS and IE to denote if they are recognized by Netscape and Internet Explorer.

Table 5.5.2 Events Supported by Javascript

Event	*IE NS*	*Target*	*Cause*	*Properties*
Abort	NS IE	image	The user aborts the loading of the image by clicking another link or the stop button	
Blur	IE	[illegible]	[illegible]	

```
<h5> This is a sample page to illustrate OnHelp... Press F1 key for help
<br><br> </h5>
<body>
<h5>Type the city code </h5>
<input type="text" name="Citycode"
onHelp="alert('Citycode Chennai-CH \n Delhi-DL \n Mumbai-MB \n')">
</body>
</html>
```

The output is shown in Figure 5.19.

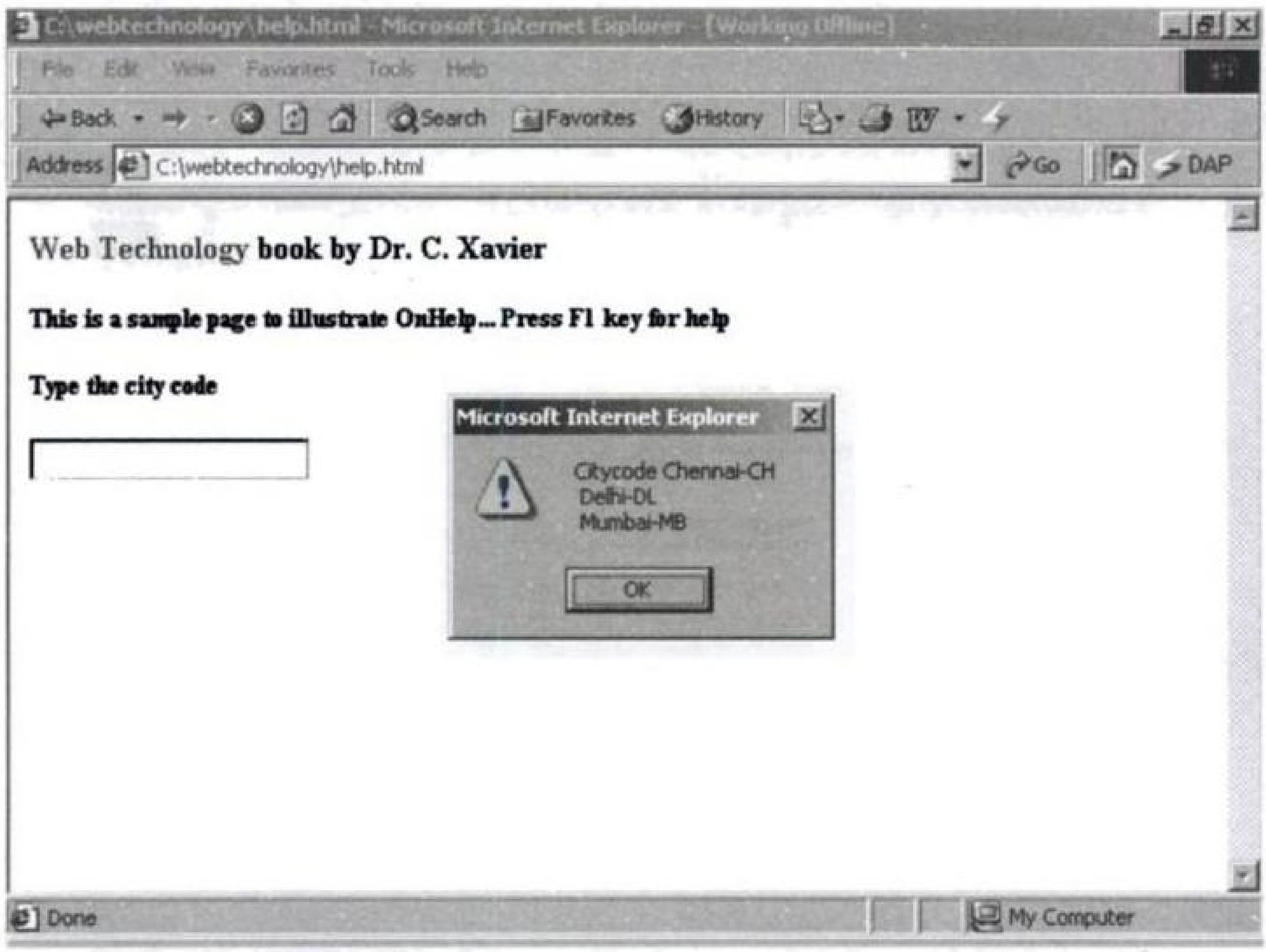


Figure 5.19 On Help Event

5.5.4 Key Press Event

A `keypress` event occurs when the user continues pressing a key. Let us see an example to illustrate this event.

```
<html>
<h4>Web Technology book by Dr. C. Xavier<br>
```

Exercises

5.1 In the form mentioned in exercise 3.1, include the Javascript to validate the data as follows:

- Name should have 25 characters and it doesn't have any numbers or special characters
- Address should have 200 characters.
- Credit card number is a 16-digit integer.

5.2 In the form mentioned in exercise 3.2, include a Java script to validate the date with month for example, September 31 is invalid?

5.3 In the form mentioned in exercise 3.3, include suitable data validation

5.4 In the form mentioned in exercise 3.4, to reserve the railway ticket add the following data validations using Javascript

- From city and To city are two different cities
- Age of passengers should not be greater than 100
- Name of the passenger should be String (no numeric digits) of maximum length 20.

5.5 In the form explained in exercise 3.5, include the following data validations using Javascript.

- The register number should be a 9-digit number.

```
<h1 align="center"><marquee> Factorial </marquee></h1>
<pre>
<center>
Enter the number : <input type=text name=a>
<br>
The Factorial is : <input type=text name=b>
<br><br>
<input type=button value="Find the Factorial"
onClick="calc(this.form)">
<br><br><hr><br><b>
</pre>
</form>
</body>
</html>
```

The Output is shown in Figure 6.2.

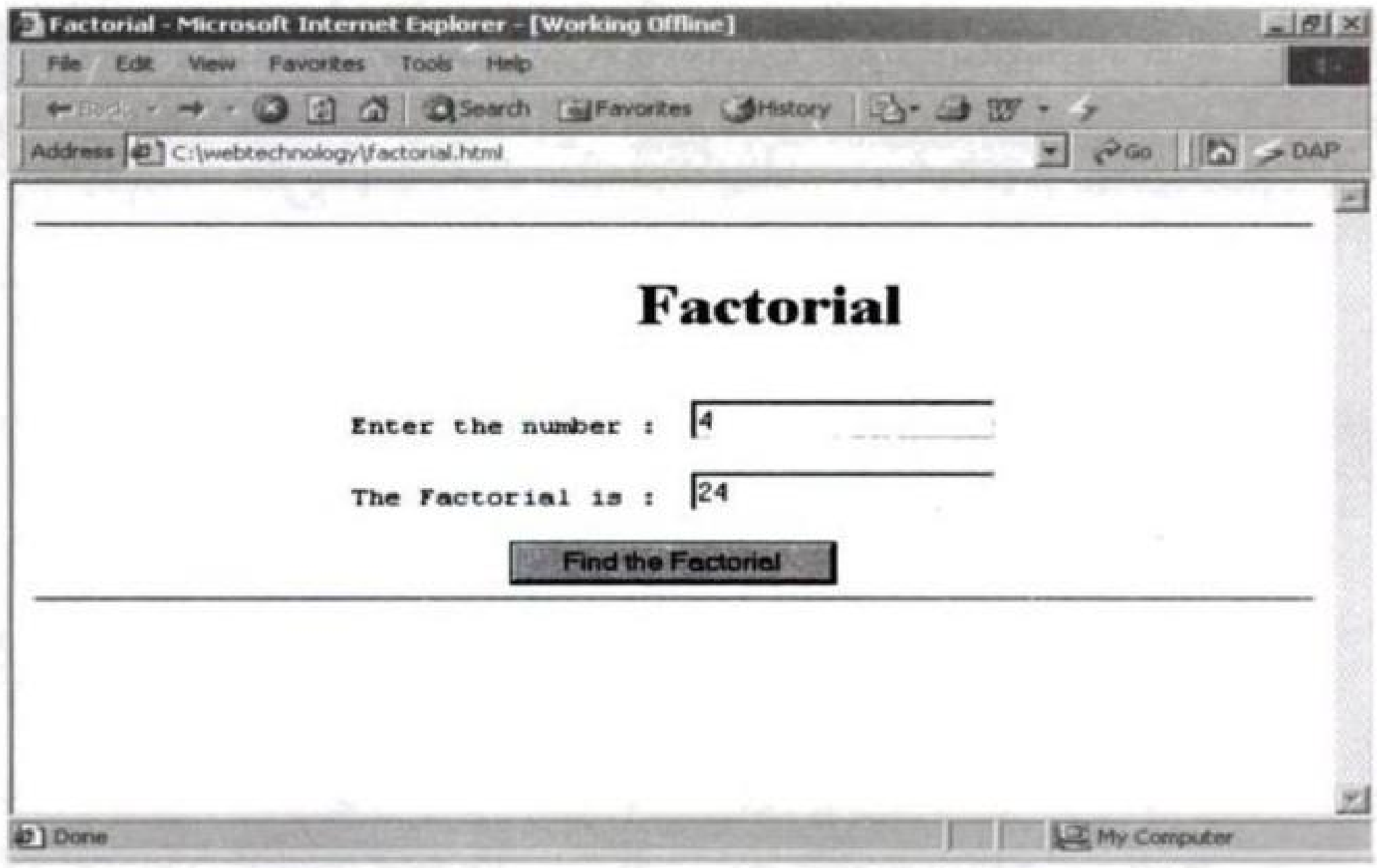


Figure 6.2 Factorial Output

6.4 Simple Interest Example

This example illustrates the reading of input values and processing them. We use the calculation of simple interest for this. The simple interest for a principal value p at the rate of r% for a period of n years is evaluated by the formula.

$$\text{Interest} = p * n * r / 100$$

6.5 Income Tax Example

This example illustrates the reading of input values and processing them. We use the calculation of Income tax for this. The deduction (ded) for the annual income(anninc) is evaluated by the formula.

```
ded = anninc/3
```

The Total income (TInc) is calculated as per the following formula

```
 if(ded > 12000)
   ded=12000;
   TInc=anninc-ded;
```

The Income Tax (tax) is evaluated as follows.

```
if(TInc<=18000)
  tax=0;
if((TInc>18000) && (TInc<=25000))
  tax=0.25*(TInc-18000);
if((TInc>25000) && (TInc<=50000))
  tax=1750+0.3*(TInc-25000);
if((TInc>50000) && (TInc<=100000))
  tax=9250+0.4*(TInc-50000);
if(TInc>100000)
  tax=29250+0.5*(TInc-100000);
```

We use the form shown in Figure 6.5 for this problem.

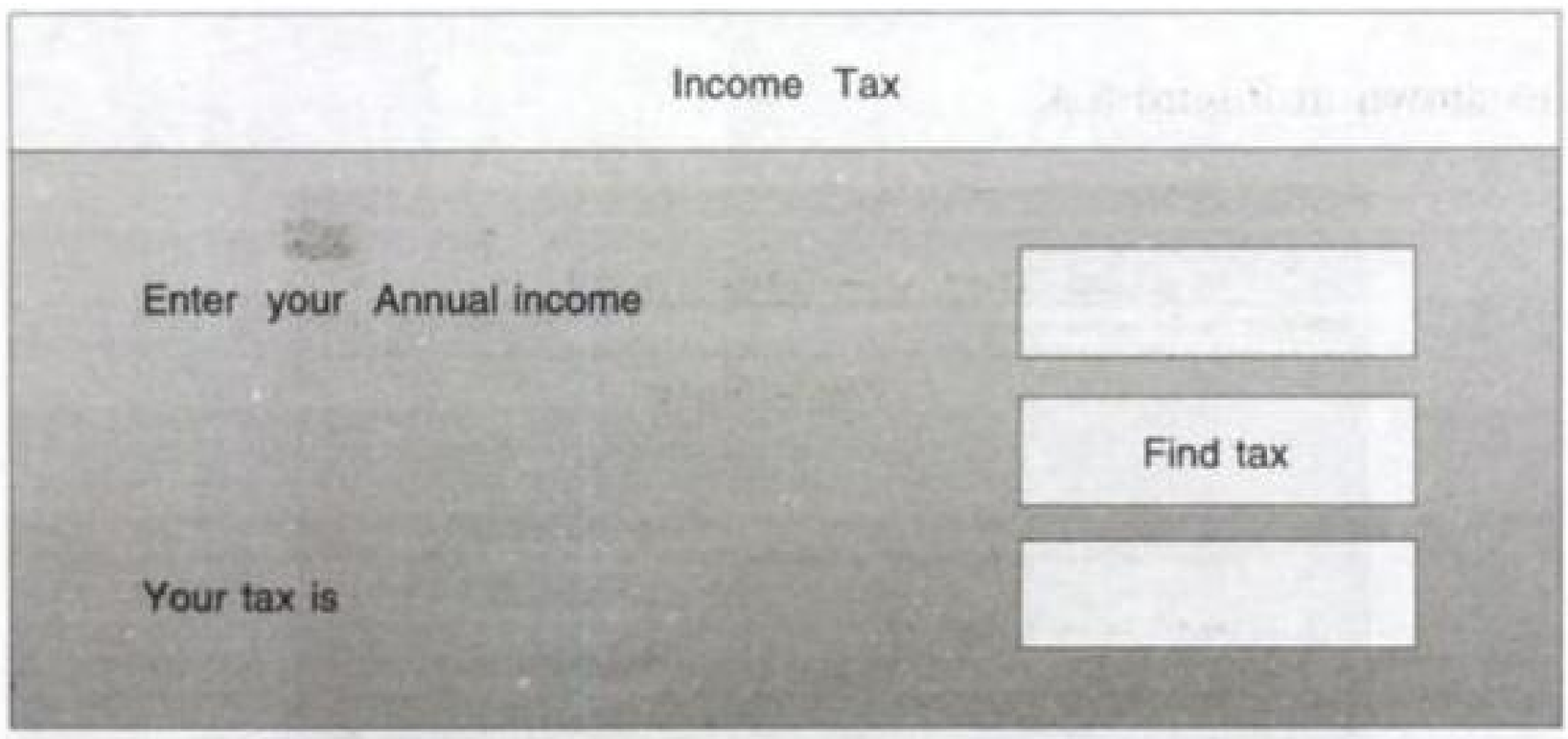


Figure 6.5 A Screen for Income Tax

The user types the Annual income on the first text input box and clicks on the "Find tax" button. Now the Javascript method findIT() is called. This method extracts data from the input box, evaluates the Income tax and displays in the text field at the bottom.

6.7.3 Form Design

We design the form with four properties radius , area , circumference and the button. The radius input text field is created as follows:

```
<input type=text name=radius onKeypress=check(this.form)>
```

Whenever we press a key in this field , the method `check(this.form)` is called.

Similarly the area and circumference text fields are created . When we click the button the method `calculate(this.form)` is called.

```
<input type=text value=Calculate
onClick=calculate(this.form)>
```

The complete HTML code is given below:

```
<html>
<head>
<title>CIRCLE</title>
<script>
  var f,a;
  function check(f)
  {
    var a=window.event.keyCode;
    if(!(a>=48 && a<=57))
    alert("Type a number between 0 & 9");
  }
  function circle(a)
  {
    this.radius=a;
    this.area=Math.PI*a*a;
    this.circumference=2*Math.PI*a;
  }
  function calculate(f)
  {
    var c=new circle(f.radius.value);
    f.area.value=c.area;
    f.circumference.value=c.circumference;
  }
</script>
</head>
<body bgcolor=Pink>
<form>
<br><hr><i><b><h2><marquee>Circle</marquee></h2></b></i>
<hr><br><pre><center>
Enter the radius and press the calculate button to get area and
circumference
```

In the above code , the ASCII code of the Keystroke is captured in a variable *a* as follows:

```
Var a = window.event.KeyCode
```

Now we must verify if a is equal to the ASCII code of 0,1,2,...or 9. That is the value of a must be between 48 and 57 (**Note:** the ASCII code of 0 is 48 that of 1 is 49 , ... ASCII of 9 is 57).

6.8.3 Form Design

We design the form with four properties. They are three coefficients of equation and the button. The coefficients of equation are the input text fields they are created as follows:

```
  A <input type=text name=a value=0
onkeypress=check(this.form)>
  B <input type=text name=b value=0
onkeypress=check(this.form)>
  C <input type=text name=c value=0
onkeypress=check(this.form)>
```

Whenever we press a key in this field , the method check(this.form) is called.

When we click the button the method calcroot(this.form) is called.

```
<input type=button value=Solve onClick=calcroot(this.form)>
```

The complete HTML code is given below:

```
<html>
<head><title>Quadratic Equation</title>
<script>
var a,b,c,f;
function check(f)
{
  var a=window.event.keyCode;
  if(!(48<= a && a<= 57))
  alert("Type a number between 0 & 9");
}
function calcroot(f)
{
  a = parseFloat(f.a.value);
  b = parseFloat(f.b.value);
  c = parseFloat(f.c.value);
  if(a==0)
    alert("This is not a quadratic Equation. Type a non-
    zero value for A");
else
```

Exercises

6.1 Write a HTML page with Javascript to find the area of a rectangle.

6.2 Write a HTML page with Javascript to find the present value of a Car. Accept necessary input and evaluate as per a standard formula.

6.3 Write a HTML document with Javascript to find the age of a person in completed years.

6.4 Write a HTML document with Javascript to count the number of Vowels in a text typed in a Text Area.

6.5 Write a HTML document to input two dates and find the duration (number of days) between them. The date, month and year are input in three select boxes.

5. The next step is to generate a special Servlet from the JSP file. All the HTML required is converted to `println` statements.
6. The Servlet source code is compiled into a class.
7. The Servlet is instantiated, calling the `init` and `service` methods.
8. HTML from the Servlet output is sent via the Internet.
9. HTML results are displayed on the user's web browser.

7.5 JSP Servers

There are many JSP capable web-servers available, most of which can be downloaded for free evaluation and/or development. Some of them are:

- Java Web Server
- WebLogic from BEA
- WebSphere from IBM
- Apache Web Server
- Blazix
- ServletExec
- JRun

7.5.1 Booting the Web Server

The web server must be first started (booted) to get the JSP executed. In the case of Java Web Server 2.0 (used in preparation of this book), the web server is booted as follows:

- Open the command Prompt and go to the bin folder of the web server. C:\Java Web Server 2.0\bin.
- Type the command **`httpd`** and press enter

This is illustrated in Figure 7.3.

```
out.println(d);
%>
</BODY>
</HTML>
```

We get the same output as the previous one. Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type javax.servlet.jsp.JspWriter.

7.6 JSP Tags

JSP supports four main tags. They are

1. Declaration tag
2. Expression tag
3. Directive tag
4. Scriptlet tag
5. Action tag

7.6.1 Declaration tag (<%! %>)

This tag allows the developer to declare variables or methods. Before the declaration you must have <%! . At the end of the declaration, the developer must have %>. Code placed in this tag must end in a semicolon (;). Declarations do not generate output so are used with JSP expressions or scriptlets.

Example

```
<%!
private int counter = 0 ;
private String getAccount( int accountNo) ;
%>
```

7.6.2 Expression Tag (<%= %>)

This tag allows the developer to embed any Java expression and is short for out.println (). A semicolon (;) does not appear at the end of the code inside the tag. For example, to show the current date and time.

```
Date: <%= new java.util.Date () %>
```

View this JSP in your browser, and you will see your original `hello.jsp` get included in the new JSP.

7.6.4 Scriptlet Tag (<% ... %>)

Between <% and %> tags, any valid Java code is called a *Scriptlet*. This code can access any variable or bean declared.

For example, to print a variable.

```
<% String book = "Web Technology by Dr. C. Xavier " ;
out.println ( book ) ;
%>
```

7.6.5 Action Tag

There are three main roles of action tags:

1. enable the use of server side Javabeans
2. transfer control between pages
3. browser independent support for applets

So far we know that the developer can create *Javabeans* and interact with Java objects. There are several objects that are automatically available in JSP called implicit objects.

The implicit objects are shown in Table 7.2.

Table 7.2 Objects

Variable	*Of Type*
Request	Javax.servlet.http.httpservletrequest
Response	Javax.servlet.http. httpservletresponse
Out	Javax.servlet.jsp.JspWriter
Session	Javax.servlet.http.httpsession
PageContent	Javax.servlet.jsp.pagecontext
Application	Javax.servlet.http.ServletContext
Config	Javax.servlet.http.ServletConfig
Page	Java.lang.Object

Page object represents the JSP page and is used to call any methods defined by the servlet class. Config object stores the Servlet configuration data.

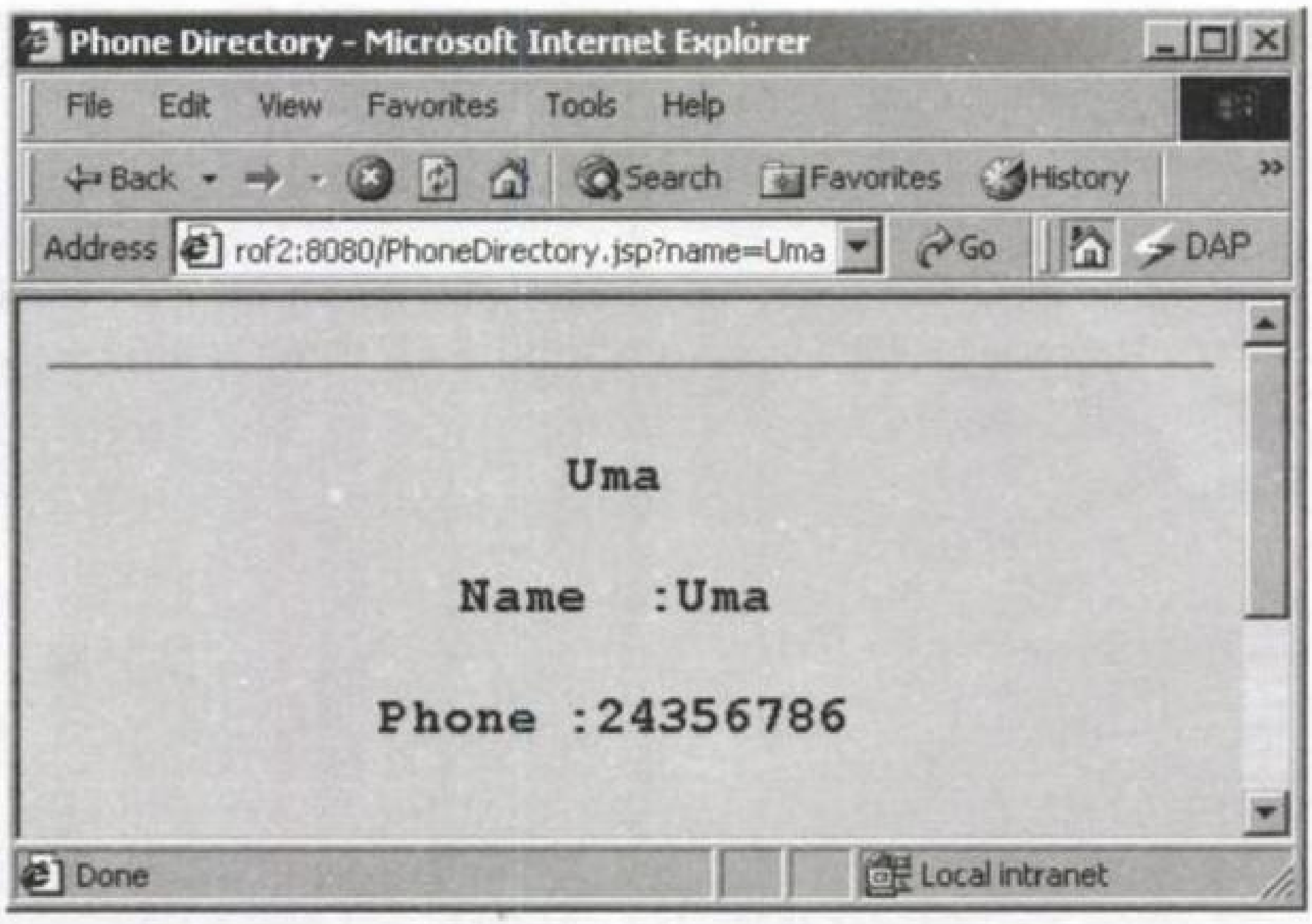


Figure 7.10 Phone Directory

Example 7.7.3 The getRemoteHost()Method of Request Object

Consider the following:

```
request.getRemoteHost()
```

This will give the details of the client from where the request has come..

For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.).

```
out.println( request.getRemoteHost());
```

Example 7.7.4 The getQueryString() Method of Request Object

Scriptlets have access to the same automatically defined variables as expressions. So, for example, if you want output to appear in the resultant page, you would use the `out` variable.

```
<%
String queryData = request.getQueryString();
out.println("Attached GET data: " + queryData);
%>
```

Note that code inside a scriptlet gets inserted *exactly* as written, and any static HTML (template text) before or after a scriptlet gets converted to `print` statements. This means that scriptlets need not contain complete Java statements, and blocks left open can affect the static HTML outside of the scriptlets. This is illustrated in the following Example 7.7.5.

CHAPTER 8

Business Processing with JSP

In this chapter we illustrate the division of the work between client and server. As we have already seen, the client side JavaScript code takes care of the preservation and data validation. The server side JSP code implements the business process. This is illustrated using a simple problem that was already explained in section 6.3.

Example 8.1 Factorial

In section 6.3 we developed a HTML and JavaScript document to evaluate the factorial of a number. This problem has the following modules:

1. A HTML form is shown as per the design shown in Figure 6.1. the user types the number n and clicks the button "Find Factorial".
2. The Client(browser) validates the data n.
3. The Client sends a request to the server. The request takes n to the server and asks for the factorial of n.
4. The Server receives the number n, evaluates the factorial of n and sends the result to the browser.

Module 1: HTML Form

The HTML form has a text field and a submit button as shown in Figure 8.1.

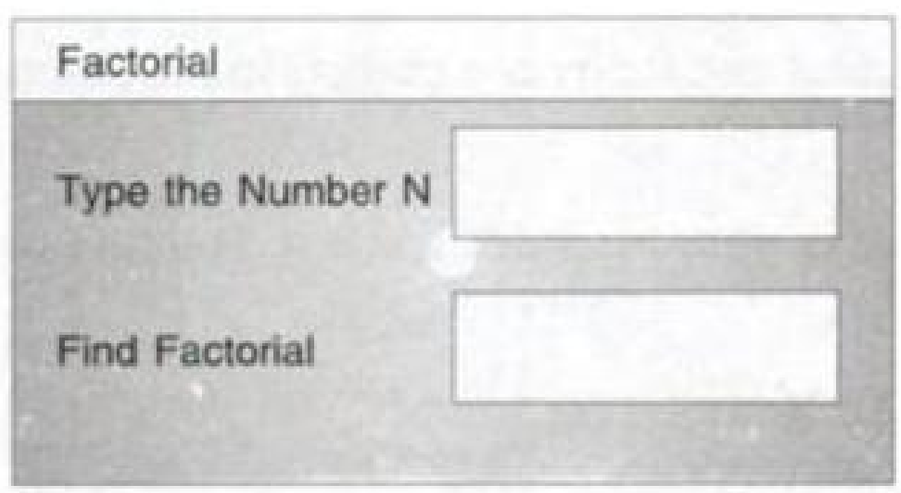


Figure 8.1 Screen to Find Factorial of n

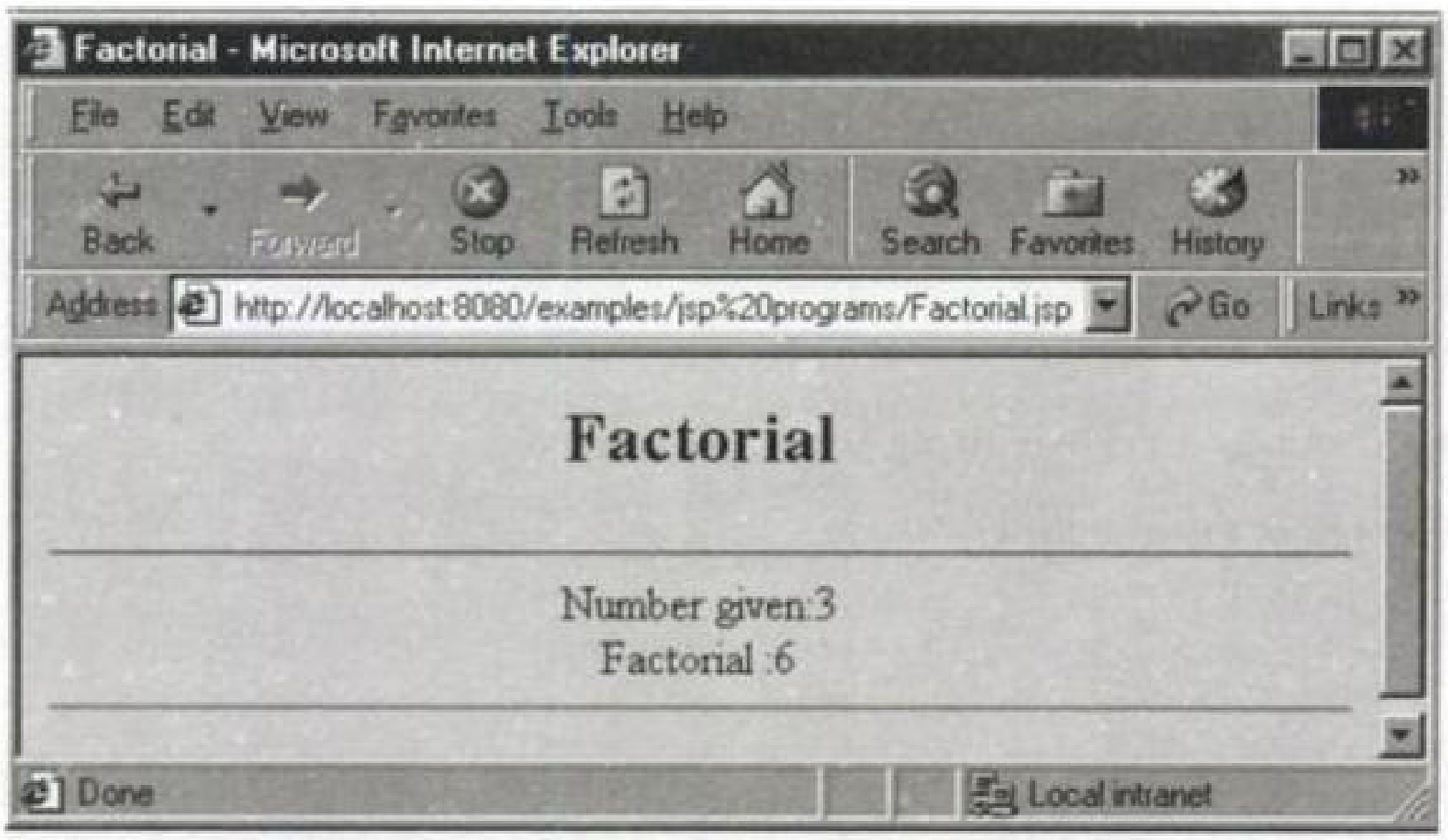


Figure 8.3 JSP Page for Factorial

Example 8.2 Simple Interest

In example 6.4 we developed a HTML and JavaScript document to evaluate the Simple Interest of a number. This problem has the following modules:

1. A HTML form is shown as per the design shown in Figure 6.3. the user types the Principal, number of years, rate of interest and clicks the button "Simple Interest".
2. The Client(browser) validates the data .
3. The Client sends a request to the server. The request takes the data to the server and asks for the Simple interest.
4. The Server receives the data, evaluates the Simple interest and sends the result to the browser.

Module 1: HTML Form

The HTML form has 3 text fields and a submit button as shown in Figure 8.4.

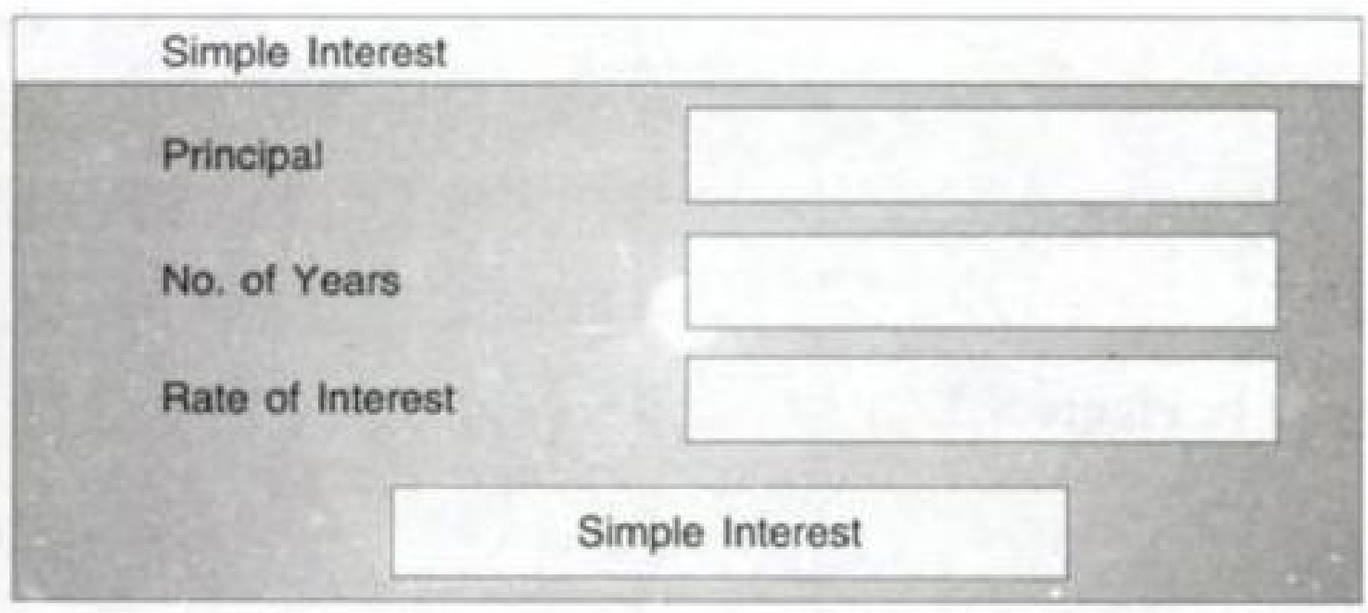


Figure 8.4 Screen to find Simple Interest

Example 8.3 Quadratic Equation

In example 6.8 we developed a HTML and JavaScript document to evaluate the Quadratic equation of a number. This problem has the following modules:

1. A HTML form is shown as per the design shown in Figure 6.11. The user types the coefficients a, b, c and clicks the button "solve".
2. The Client(browser) validates the data.
3. The Client sends a request to the server. The request takes the data to the server and asks for the roots.
4. The Server receives the data, evaluates the quadratic equation and sends the result to the browser.

Module 1: HTML Form

The HTML form has a 3 text fields and a submit button as shown in Figure 8.7.

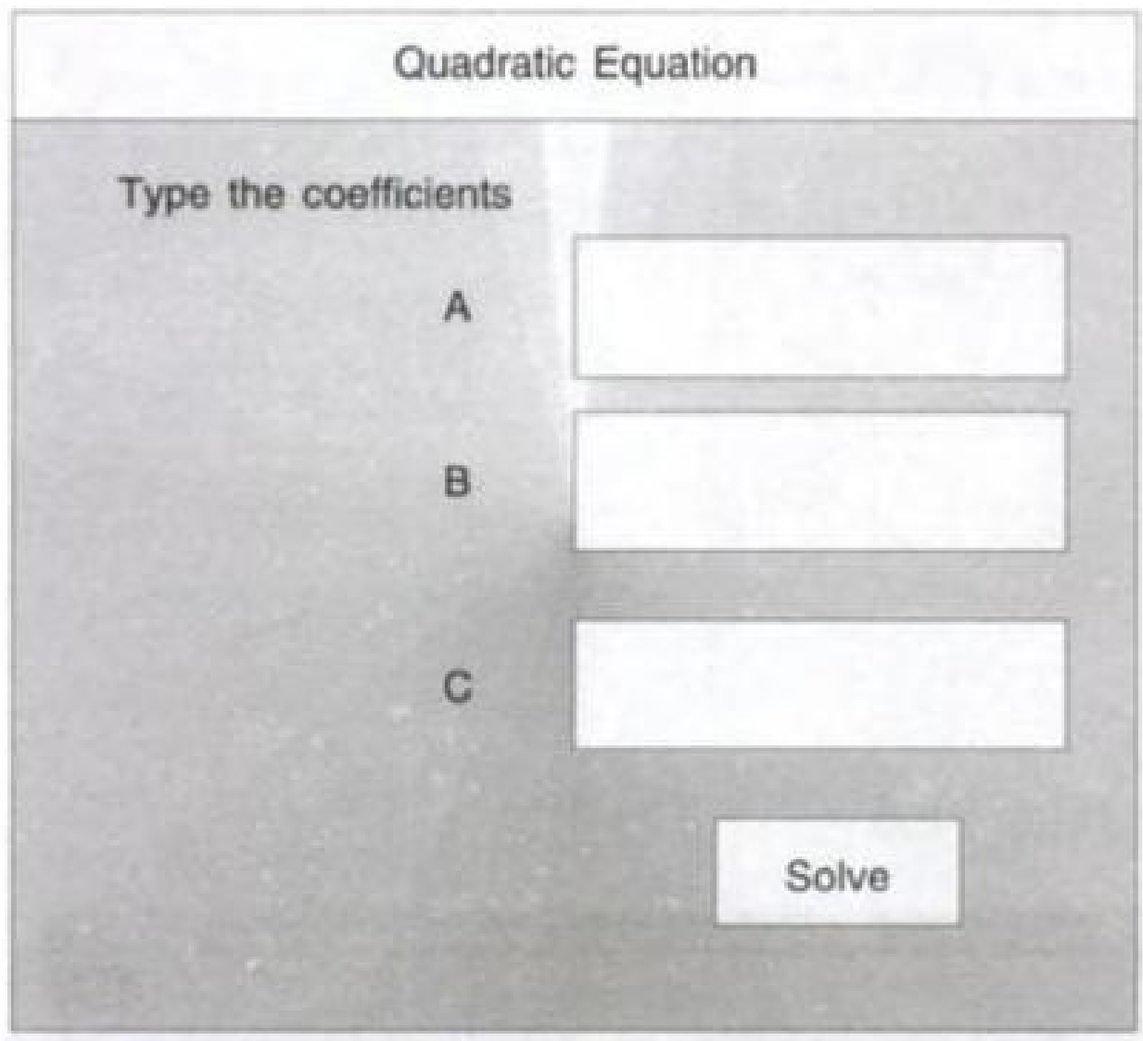


Figure 8.7 Screen for Quadratic Equation

The HTML document is shown below:

```
<html>
<head><title>Quadratic Equation</title>
<h2 align=center >Quadratic Equation</h2>
<script>
  var a,b,c,f;
  function check(f)
```

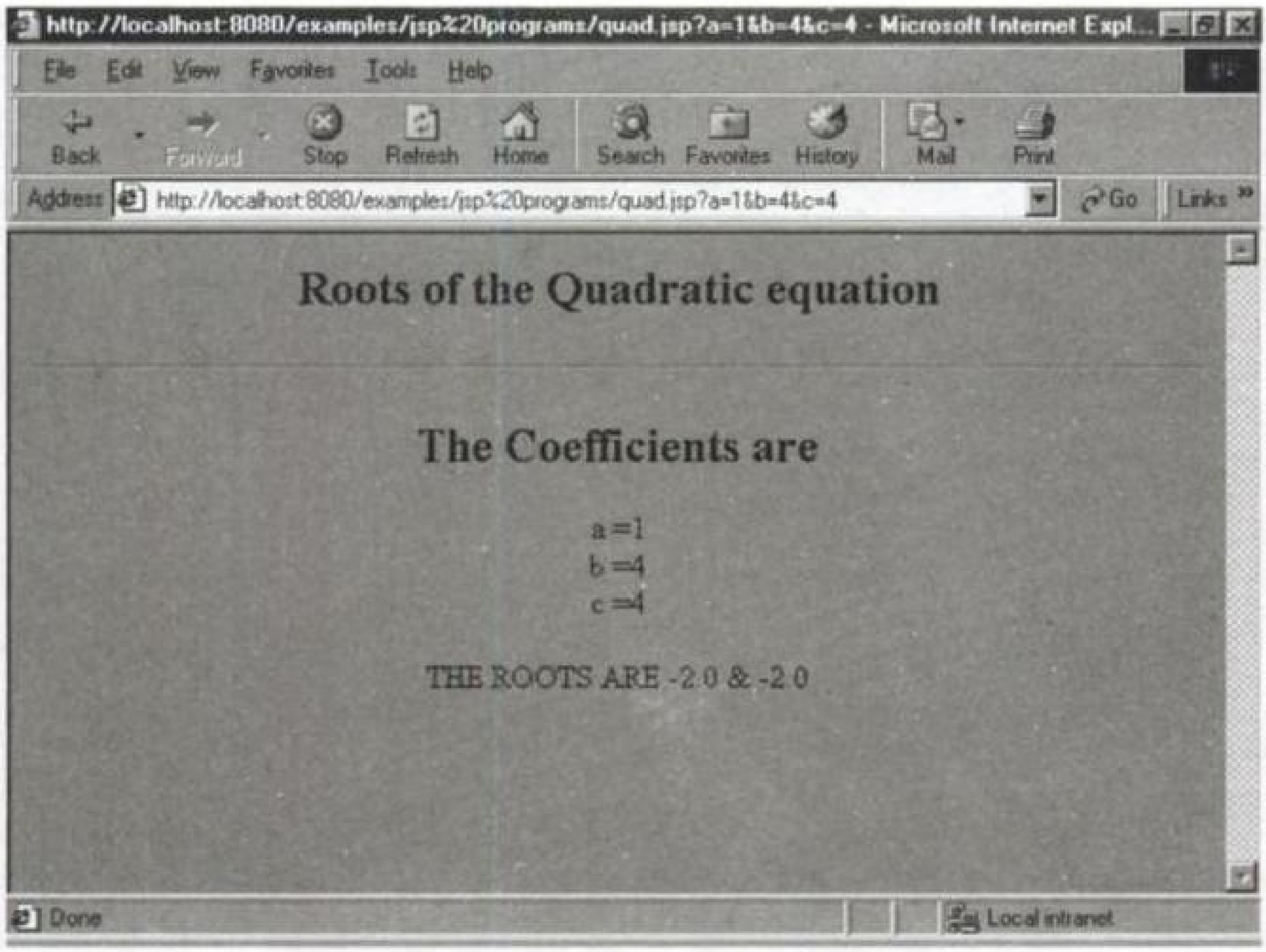


Figure 8.9 JSP Page for Quadratic Equation

Example 8.4 Income Tax

In example 6.5 we developed a HTML and JavaScript document to evaluate the Income Tax . This problem has the following modules:

1. A HTML form is shown as per the design shown in Figure 6.5. the user types the annual income and clicks the button "find tax".
2. The Client(browser) validates the data .
3. The Client sends a request to the server. The request takes the data to the server and asks for the tax.
4. The Server receives the data, evaluates the annual income and sends the result to the browser.

Module 1: HTML Form

The HTML form has a 1 text field and a submit button as shown in Figure 8.10.

```
<br>
<br>
  <%out.println("Income Tax is : " +tax);
%>
<br>
<br>
<hr>
</font>
</center>
</body>
</html>
```

The output is shown in Figure 8.12.

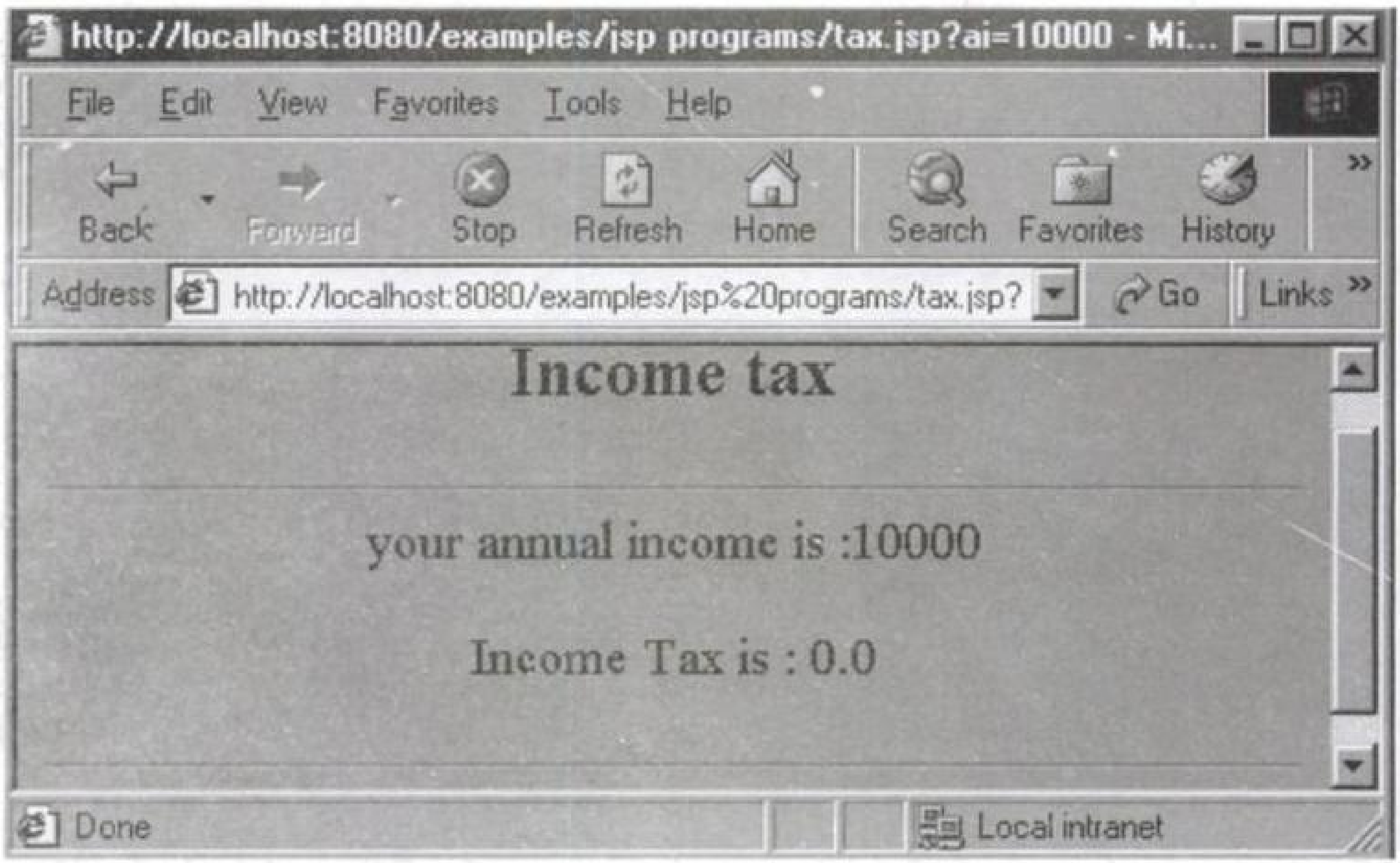


Figure 8.12 JSP Page for Income Tax

Example 8.5 Sales Commission

In example 6.6 we developed a HTML and JavaScript document to evaluate the Sales Commission . This problem has the following modules:

1. A HTML form is shown as per the design shown in Figure 6.7. the user types the total sales and clicks the button "find commission".
2. The Client(browser) validates the data .

User Profile with XML—A Web Design Project

The results of St. Xavier's College have been displayed in a web page. Access to the web page is given only to some restricted users. Define an XML document to have the user profile such as userID, Password, first name, last name and date of birth. In the HTML page, the user name and password are got from the user and verified with the userID and password of the permitted users given in XML Document. If they match, the web page showing the result of St. Xavier's College (Autonomous) is shown. Otherwise permission is denied.

14.1 Web Design

The XML Document will have the user profile of authorised viewers of the web page. In the welcome page, two text fields are used to get the userID and password of the user. When the user types the data and presses the GO button, the JavaScript is enabled. The JavaScript loads the XML document and verifies the userID and password of the user with that of authorised users (given in XML document).

In this design we present for three documents:

1. XML document.
2. Welcome Page which gets the userID and password and checks with the XML document.
3. The page in which the results of St. Xavier's College is shown

14.1.1 XML Document

The XML document `users.xml` will define a document called `<users>`, which will contain several `<user>` tags. Each user will have the following:

1. *UserID*
2. *Password*

14.1.4 Output

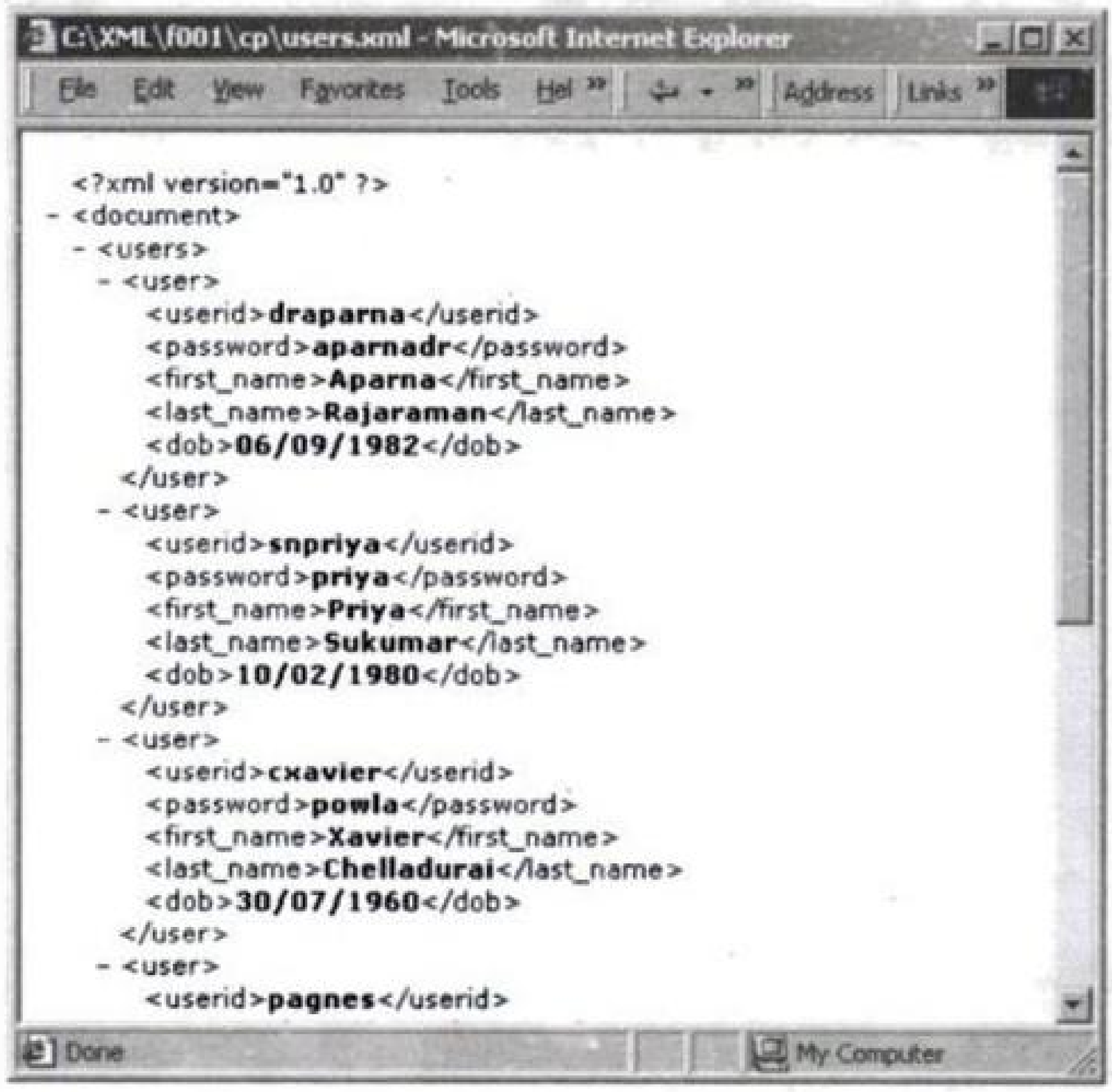


Figure 14.3 Users XML Document

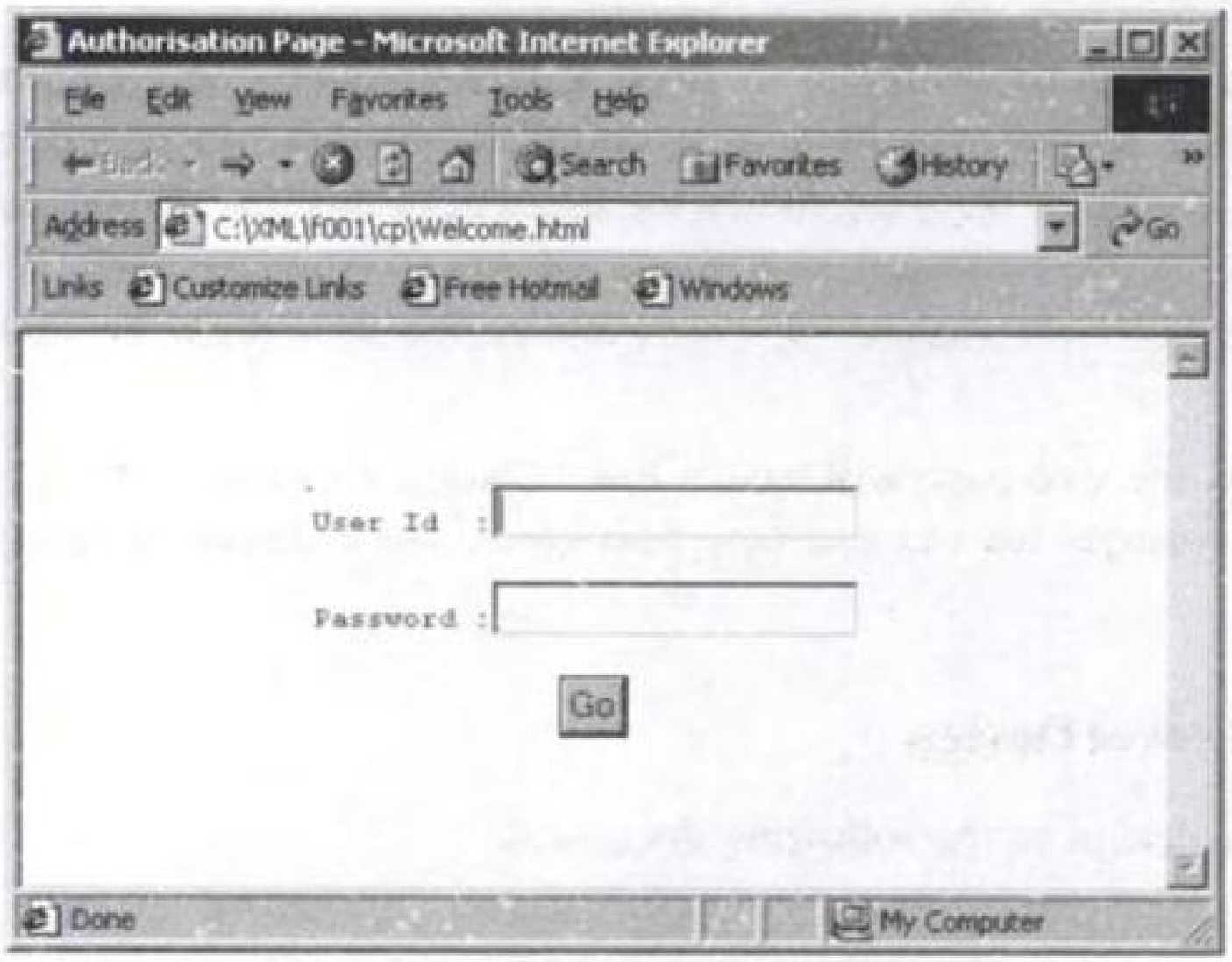


Figure 14.4 Welcome Screen

Output

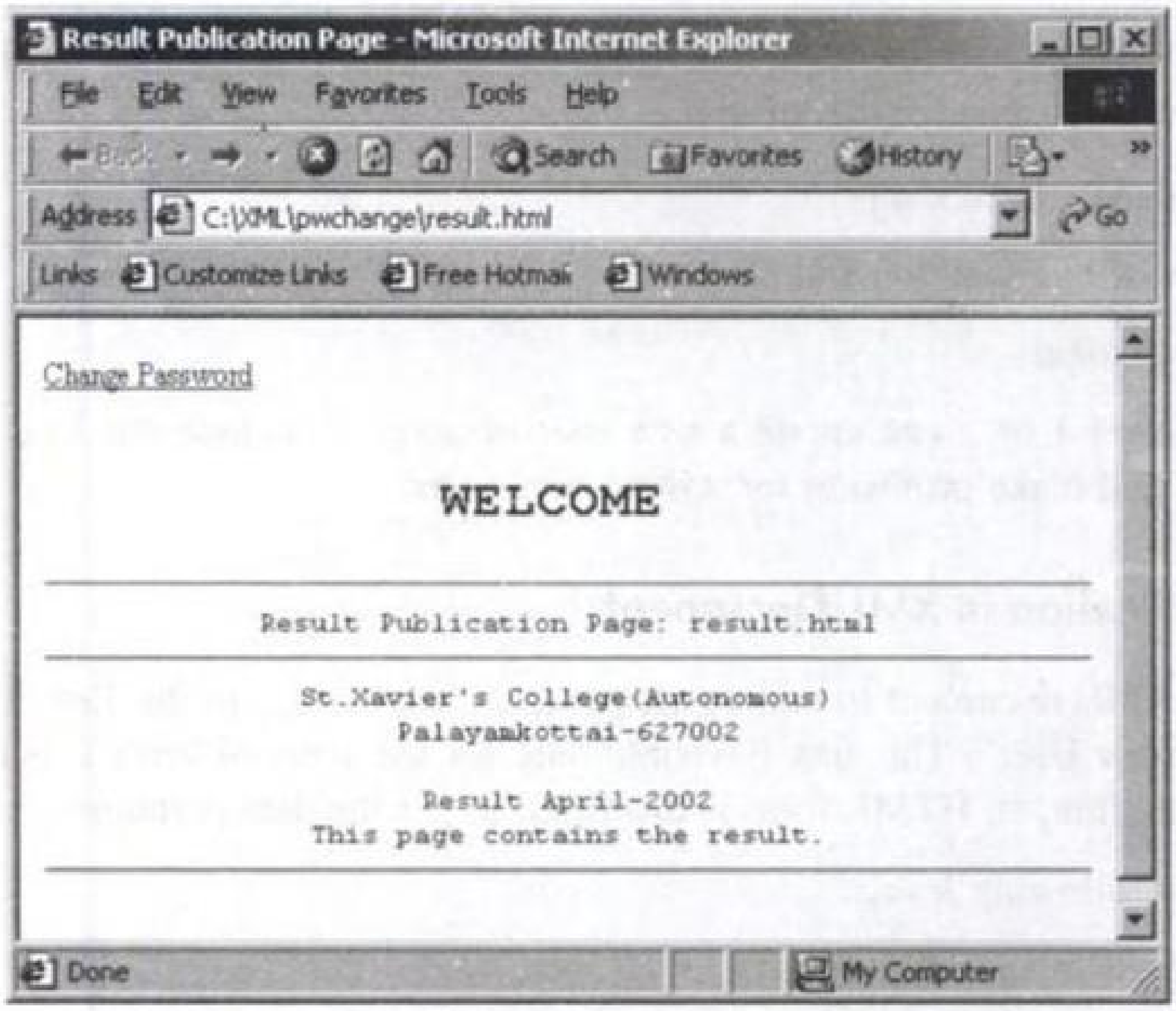


Figure 14.7 Result Page with Link to Change Password

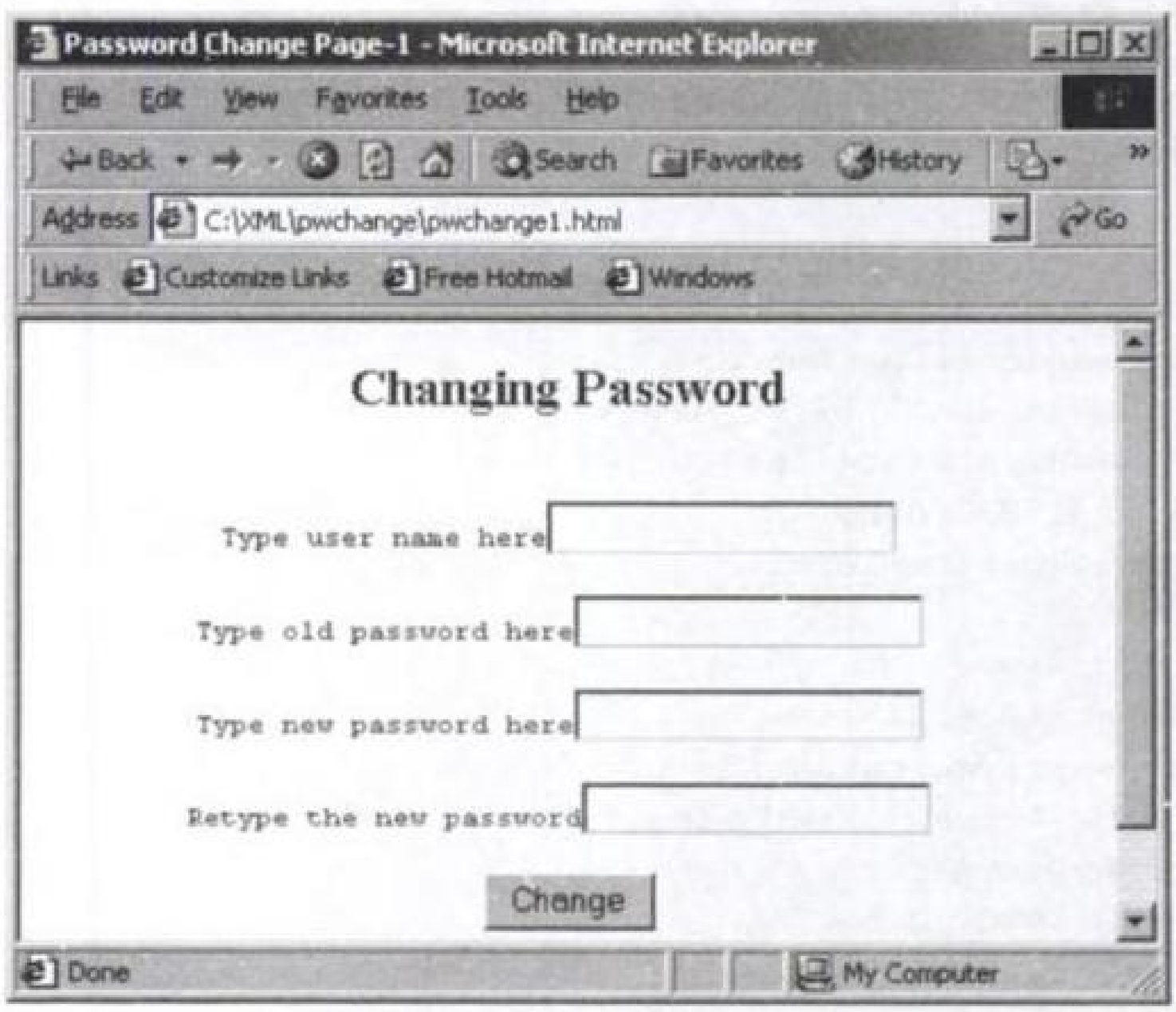


Figure 14.8 Change Password Screen

14.3.5 Output

Figure 14.10 A Link for Adding New User

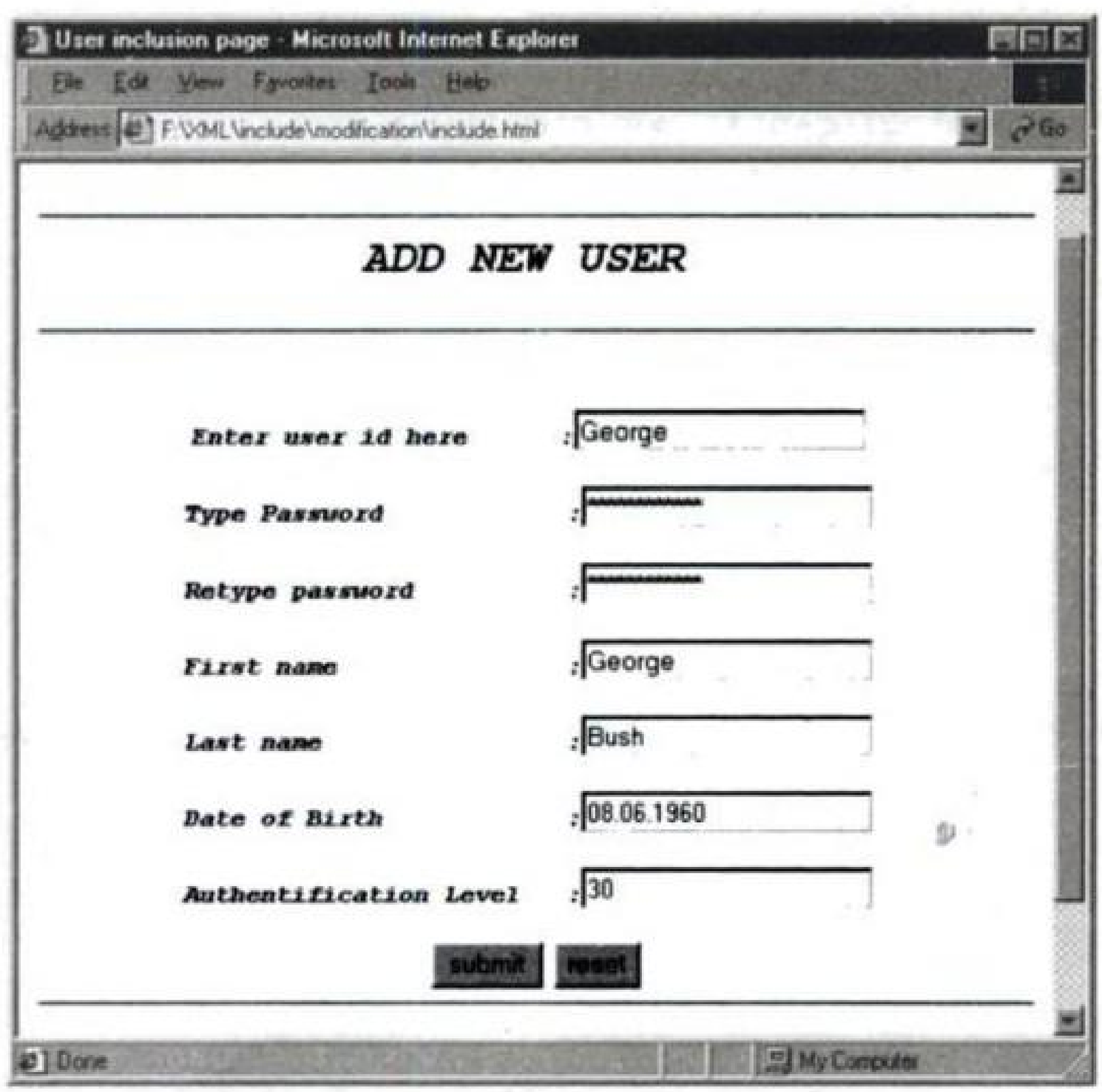
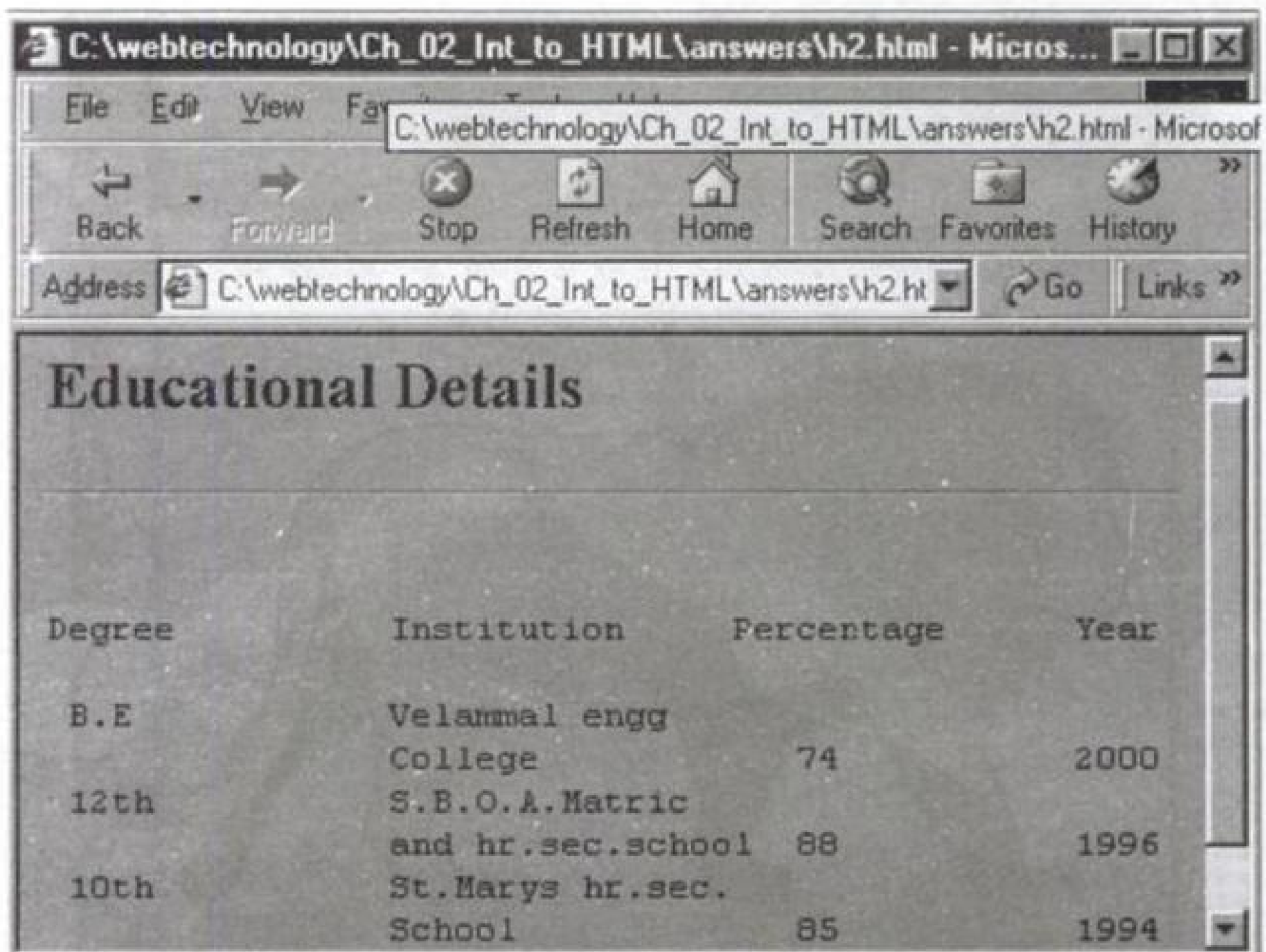
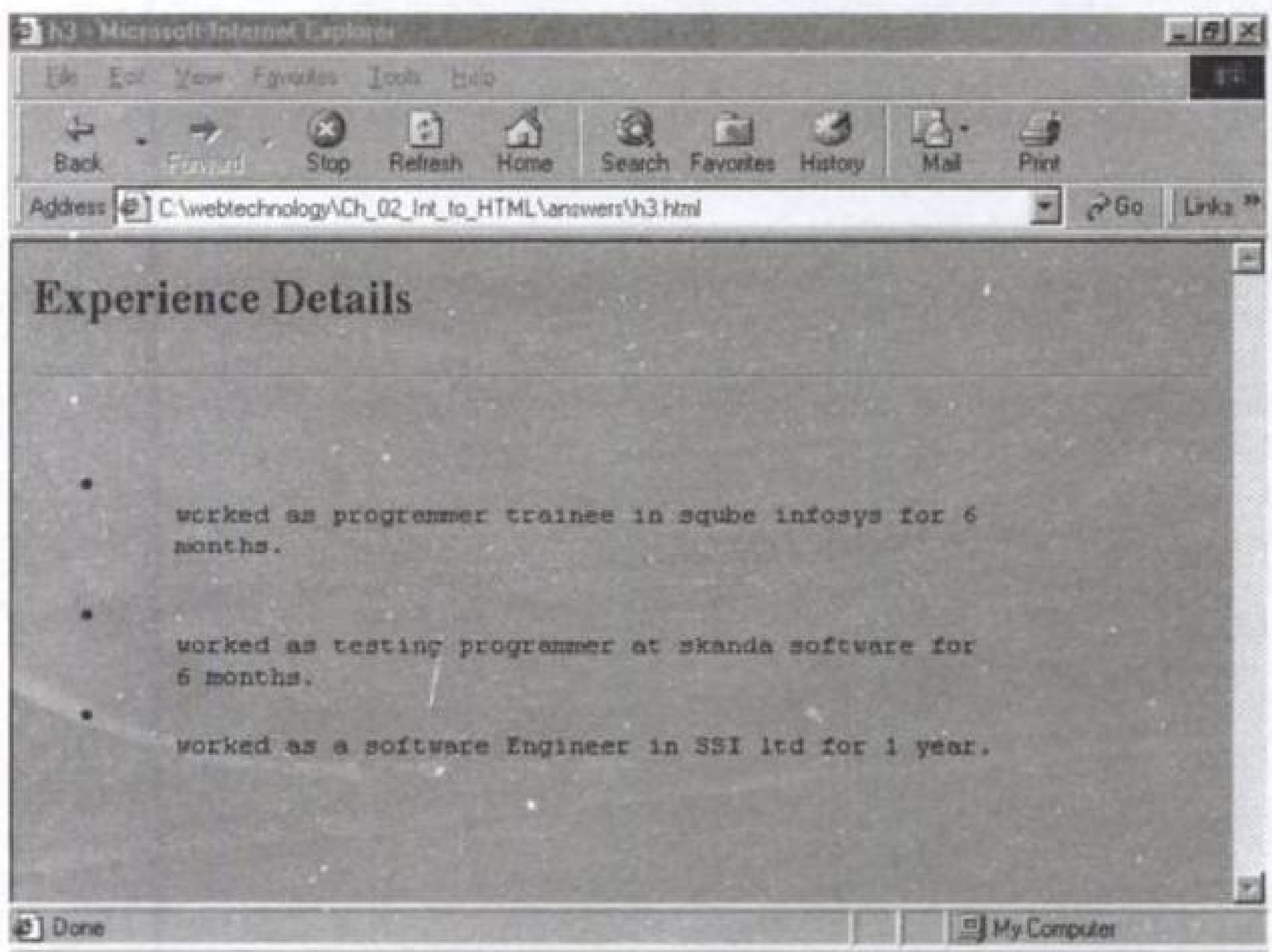


Figure 14.11 Add New User Screen

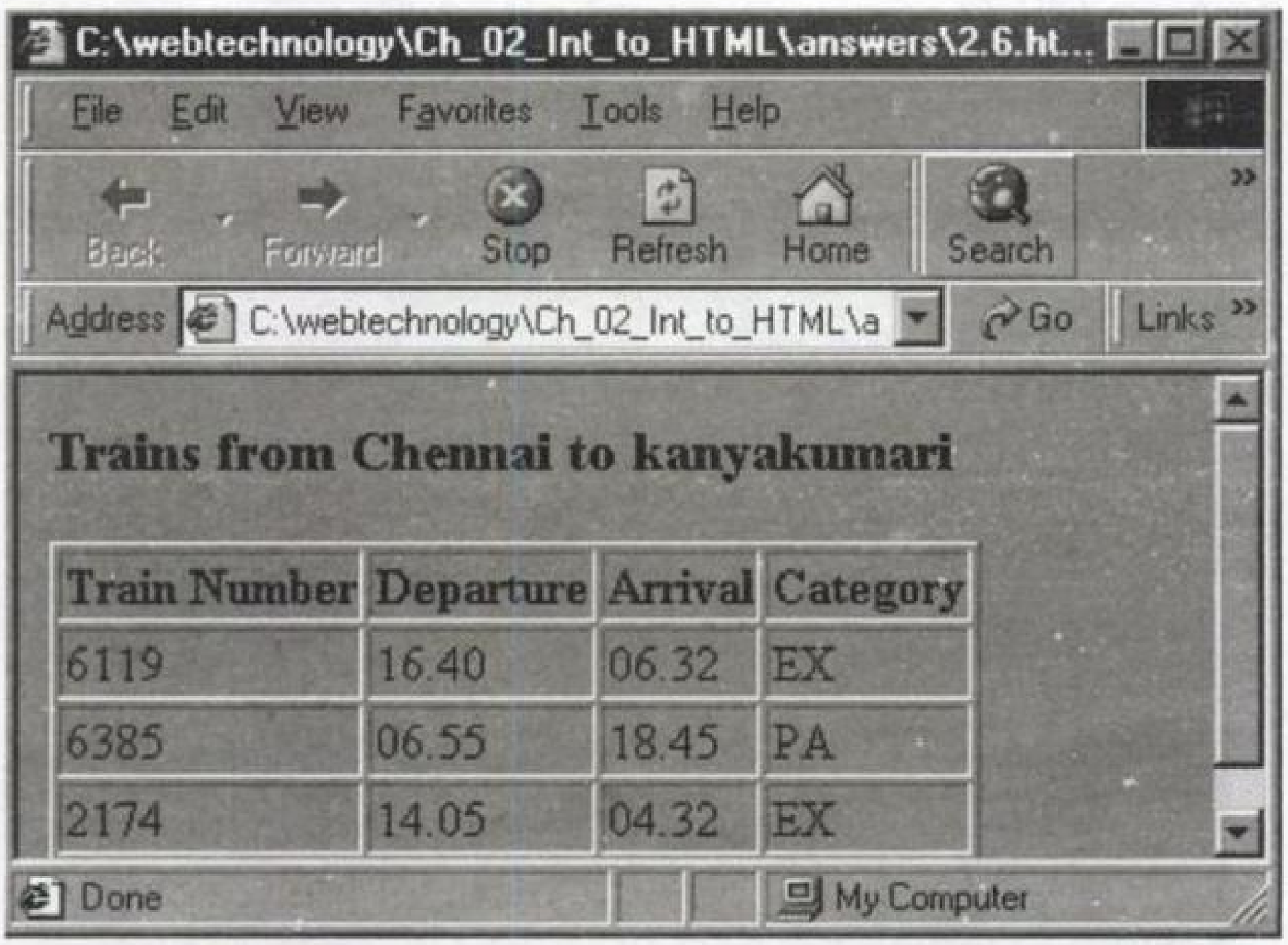




2.6 Code

```
<html>
<h3> Trains from Chennai to kanyakumari</h3>
<body bgcolor="cyan">
<table border=1 >
<tr>
<th>Train Number<th>Departure<th>Arrival<th>Category
</tr>
<tr>
<td>6119<td>16.40<td>06.32<td>EX
</tr>
<tr>
<td>6385<td>06.55<td>18.45<td>PA
</tr>
<tr>
<td>2174<td>14.05<td>04.32<td>EX
</tr>
</table>
</body>
</html>
```

The output is as follows:

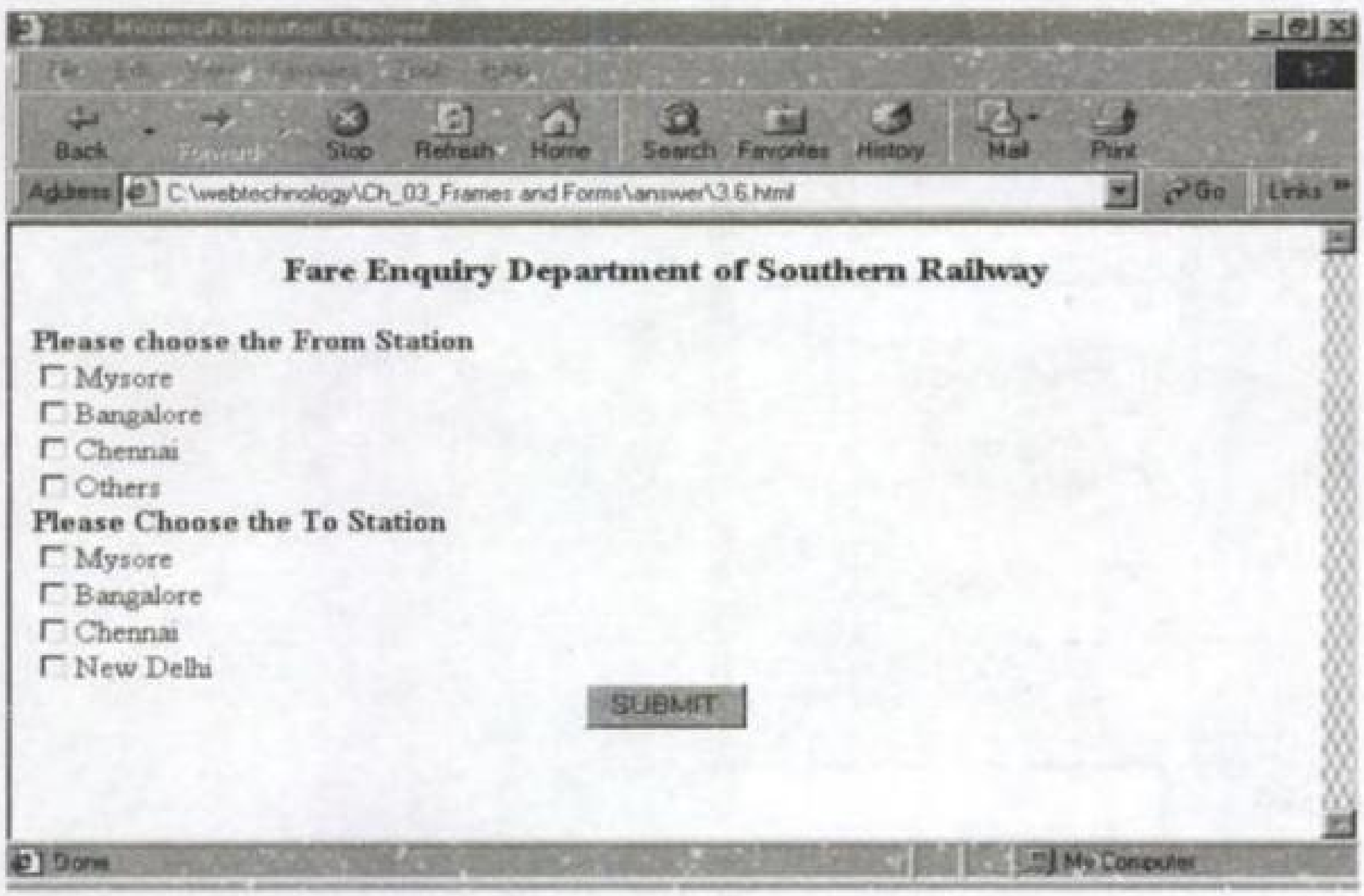


```
<option>1973
<option>1974
<option>1975
<option>1976
<option>1977
<option>1978
<option>1979
<option>1980
<option>1981
<option>1982
<option>1983
<option>1984
<option>1985
<option>1986
<option>1987
<option>1988
<option>1989
<option>1990
<option>1991
<option>1992
<option>1993
<option>1994
<option>1995
<option>1996
<option>1997
<option>1998
<option>1999
<option>2000
</select>
 </ol>
<br>
<input type="submit" value=SUBMIT>
<hr>
<br>
</center>
</body>
</html>
```

The output is as follows:

```
<body>
<h3 align="center"> Fare Enquiry Department of Southern Railway</h3>
<b>Please choose the From Station</b><br>
     <input type=checkbox name="Depart">Mysore<br>
     <input type=checkbox name="Depart">Bangalore<br>
     <input type=checkbox name="Depart">Chennai<br>
     <input type=checkbox name="Depart">Others<br>
<b>Please Choose the To Station</b><br>
     <input type=checkbox name="Arrive">Mysore<br>
     <input type=checkbox name="Depart">Bangalore<br>
     <input type=checkbox name="Depart">Chennai<br>
     <input type=checkbox name="Depart">New Delhi<br>
<center><input type="submit" value="SUBMIT"></center>
</body>
</html>
```

The output is as follows:



Chapter 4

4.1 Code

```
<html>
<head>
<title>Sum of n Numbers </title>
```

4.5 Code

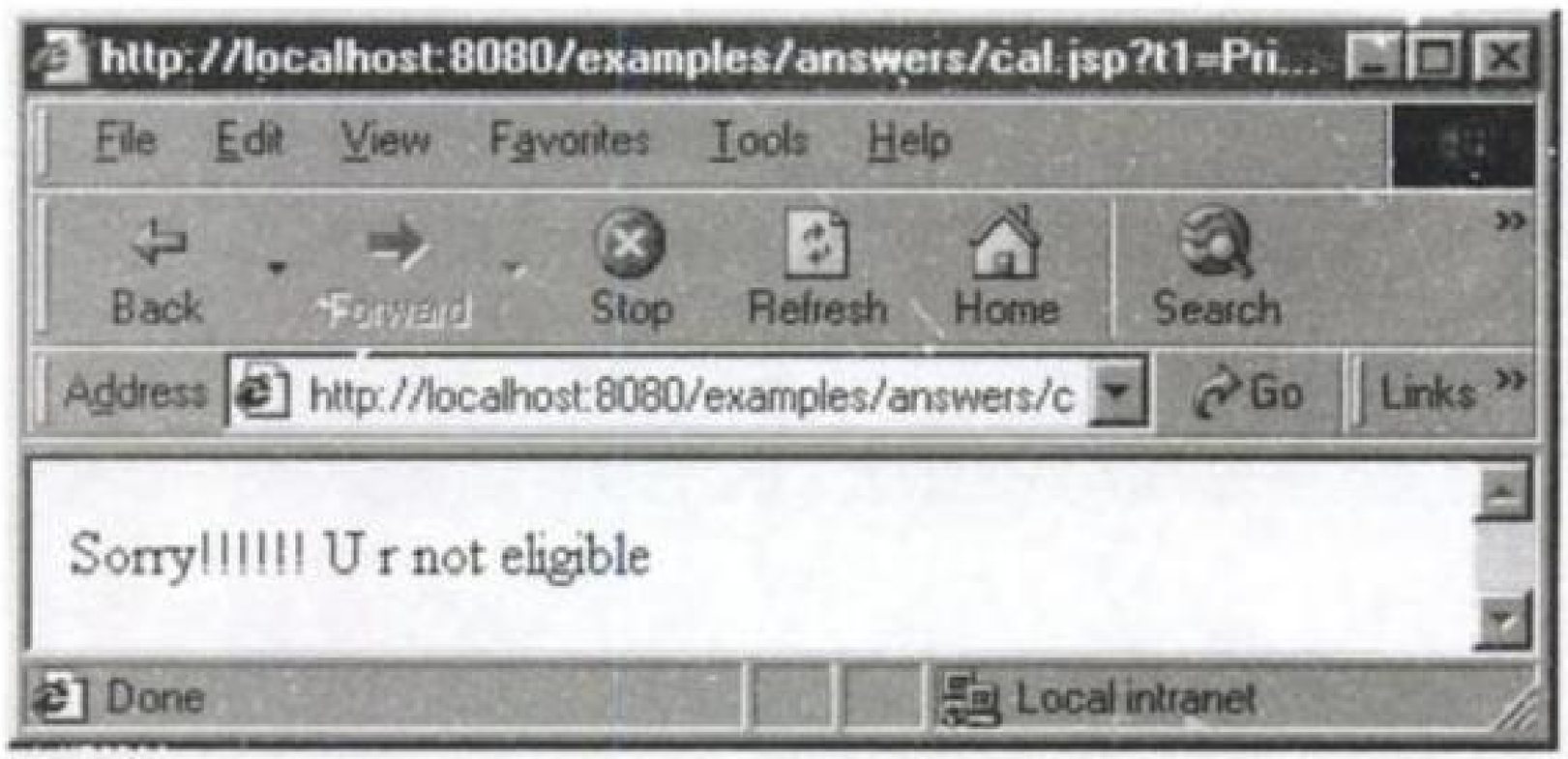
```
<html>
<head>
<title>GCD of two numbers </title>
<script>
var n,m,r,temp ;
function check(f)
{
n=parseInt(t1.value);
m=parseInt(t2.value);
if (m>n)
{
temp=n;
n=m;
m=temp;
}
do
{
r=n%m;
n=m;
m=r;
}while (r=0);
t4.value=n;
}
</script>
</head>
<body bgcolor=bisque>
<h2 align=center> Greatest Common divisor </h2>
<hr>
<pre>
Enter the first number      :<input type= text name=t1>
<br>
Enter the second number     :<input type= text name=t2
                               onBlur=check(this.form)>
<br>
The GCD of two numbers is  :<input type= text name=t4>
<br>
<hr>
</body>
</html>
```

The output is as follows:

6.2 Code

```
<html>
<head>
<title>Present value of a car</title>
<script>
function check(f)
{
var d,old,k,diff,j,i,year,date_purchase,i,p_value;
d= new Date();
year=d.getYear();
old=t1.value;
date_purchase=t2.value;
i=date_purchase.substring(6,10);
diff=year-i;
k=1;
for (j=1;j<=diff;j++)
{
k=.8*k;
}
p_value=parseInt(old*k);
t3.value=p_value;
}
</script>
</head>
<body bgcolor=bisque>
<h2 align=center > Present value of the car </h2>
<hr>
<pre>
Enter the value of the car   :<input type =text name=t1 >
<br><br>
Date of  Purchase            :<input type =text name=t2
                              onBlur=check(this.form)>
<br><br>
The present value of the car is :<input type=text name=t3>
<br><br>
<hr>
</body>
</html>
```

The output is as follows:

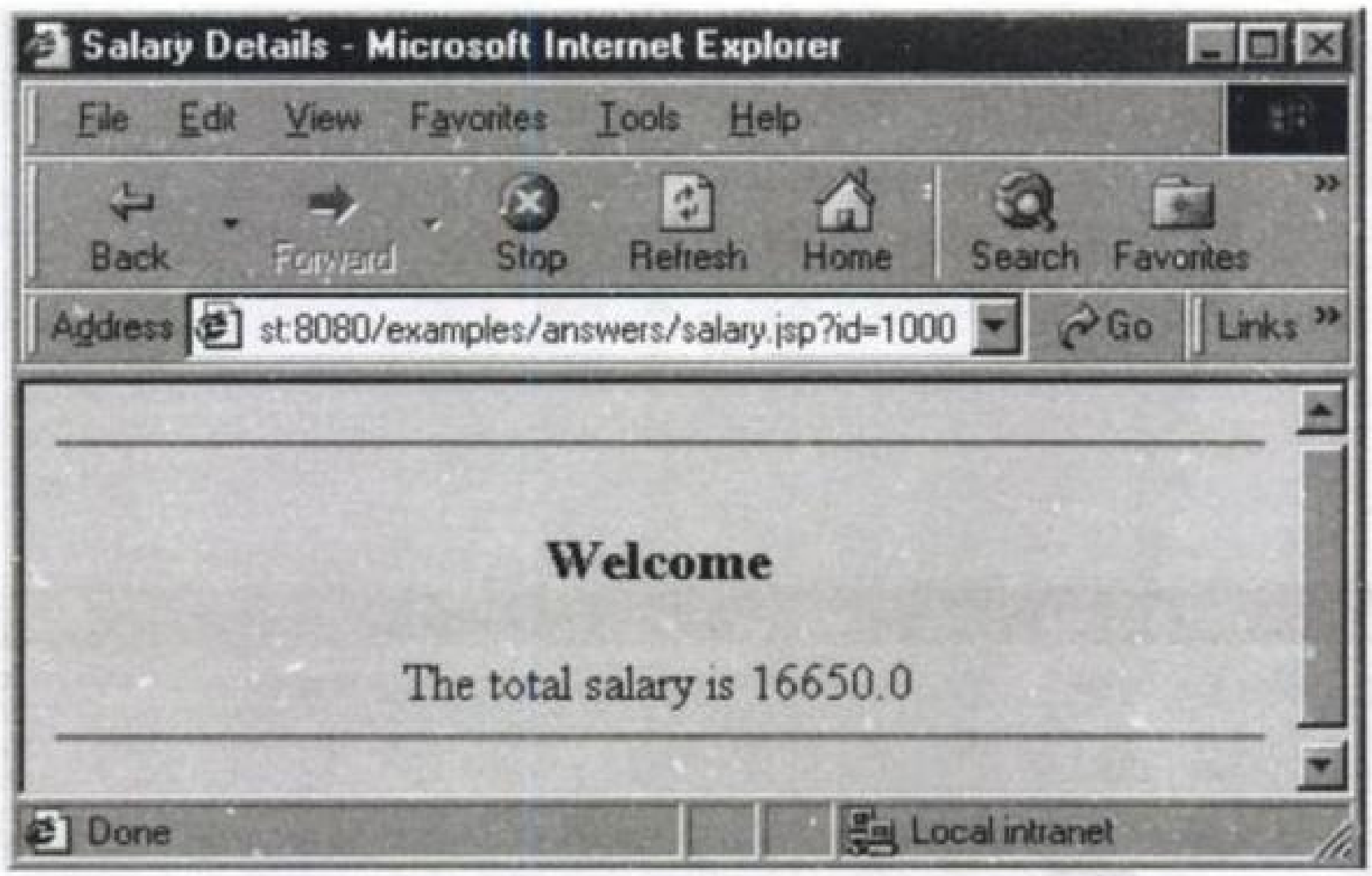


8.4 Code

```
<html>
<head>
<title>Railway Reservation System</title>
<script>
d=new Date();
document.writeln(d);
</script>
</head>
<body bgcolor=bisque>
<form action=http://localhost:8080/examples/answers/railway.jsp>
<center>
<hr>
<h2> Railway Reservation System </h2>
<br>
<hr>
<h3> Welcome Everybody !!!!!!!!</h3>
<input type=submit value=submit>
</center>
<br>
<hr>
</form>
</body>
</html>
```

JSP page railway.jsp

```
<html>
<body bgcolor=bisque>
<center>
<h2>Train Reservation Fare </h2>
```



9.3 Code

```
<html>
<head>
<title>Leave Details</title>
</head>
<body bgcolor=bisque>
<center>
<form action="http://localhost:8080/examples/answers/leave.jsp">
<h2> Leave Details </h2>
<hr>
Employee Id: <input type=text name=id>
<br><br>
Year :
<select name=year>
<option value="1995">1995 </option>
<option value="1996">1996 </option>
<option value="1997">1997 </option>
<option value="1998">1998 </option>
<option value="1999">1999 </option>
<option value="2000">2000 </option>
<option value="2001">2001 </option>
<option value="2002">2002 </option>
</select>
<br><br>
<input type =submit value=FindLeaveDetails>
```

WEB TECHNOLOGY & DESIGN

This book deals with all the technologies used in the design of services over the web. It begins with the principles and concepts used in Internet and worldwide web. HTML is explained in two chapters. Since frames and forms are vital components in interactive web pages, a separate chapter is dedicated with several examples. JavaScript, the popular scripting language used in Client side data validation is then explained with adequate object oriented style. The server side code is explained with JSP. The whole of JSP is explained and illustrated using several examples. JSP is used with JDBC for accessing databases. Java Database Connectivity is given due importance and simple web applications have been developed. Java Servlet is fully explained with several examples. Four minor projects on design and application are given in the last four chapters. These projects are fully explained according to the software development life cycle. The complete set of design documents, code and testing strategies are explained.

This book will serve as a complete textbook for various graduate and postgraduate courses.

C. Xavier is a prolific author whose computer science textbooks are used in several universities in India and abroad. His research interest includes Parallel algorithms, Data compression techniques, Image processing, Virtual reality, Object oriented software metrics and Numerical computing. He has served as Reader and Head of the Department of Computer Science, St. Xavier's College (Autonomous), Tirunelveli, (Tamilnadu). He was a member of the Technical sub-committee of IT Task Force of the Government of Tamilnadu for 3 years. He is a Project Manager in HCL Technologies Ltd., Chennai and a consultant for Convergys Corporation, USA. He is a senior life member of Computer Society of India and a member of the Project Management Institute, USA.

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS
New Delhi • Bangalore • Chennai • Guwahati • Hyderabad
Kolkata • Lucknow • Mumbai

ISBN: 81-224-1450-8