# WEB SITE OF PAKISTAN TELECOMMUNICATION

**By**

*Saima Faiz*

*&*

*Saima Siddique Alvi*

**A Project Report Submitted to**
**Quaid -i- Azam**
**University Islamabad.**

**In**

**Partial Fulfillment of the**
**Requirement of**
**Post Graduate Diploma in**
**Computer Sciences**

**Computer Center**
**Quaid –I- Azam University, Islamabad.**

**Computer Center Quaid –I- Azam University, Islamabad.**

# Final Approval

**This is to certify that we have read the project thesis submitted by Saima Siddique Alvi and Saima Faiz, and found it of sufficient standard to warrant its acceptance by the Quaid -i- Azam University, Islamabad for the Post Graduate Diploma in computer sciences.**

## COMMITTEE

**EXTERNAL EXAMINER**

**NAME** _____ **SIGNATURE:** _____

**SUPERVISOR:**

**DR. GHULAM MUHAMMAD** **SIGNATURE:** _____

**COMPUTER CENTER**

**QAU ISLAMABAD**

DIRECTOR:

**DR. GHULAM MUHAMMAD** **SIGNATURE:** _____

**COMPUTER CENTER**

**QAU ISLAMABAD**

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِيْمِ

# DEDICATION

**TO OUR PARENTS,**
**WHO ARE ALWAYS WITH US**
**TO LOVE, CARE AND ENCOURAGE.**

# ACKNOWLEDGEMENT

First and formost, we thank Allah the Beneficent and the Merciful, who always helps us to accomplish our goals in life.

Our great appreciation and thanks to **Dr. GHULAM MUHAMMAD** for giving us an opportunity to take such a challenging task. He deserves our appreciation that contributed a lot making our project successful and meaningful.

Our heartiest gratitude and thanks are to **Mr. JAVED HUSSAIN** for his enduring support and guidance throughout the project.

We are thankful to our family for their support, which is always necessary for completion of our goals.

We are thankful to all our friends and class fellows for their working support and the creative ideas, which enhanced the beauty of our project and especially to our loving brother **Ahmad Naseem Alvi** for his cooperation and encouragement throughout our project.

# *TABLE OF CONTENTS*

- **The cache and toolbars**

## 5. HYPER TEXT MARKUP LANGUAGE (HTML)

- **Introduction To HTML**
- **History of Hypertext**
- **The Best Online Help Programs**
- *History of HTML*
- *What is HTML?*
- *HTML Tags*
- *Headings*
- *Paragraph*
- *Line Break*
- *Block quote*
- *Lists*
- *Hyperlink*
- *Getting the Right Format*

## 6. JAVA SCRIPTS

- **What is JavaScript?**
- **How Does it Work?**
- **History of JavaScript**
- **Live Script**
- **Java**
- **Plateforms & browsers**
- **What can you do with JavaScript?**
- **JavaScript Reserved Words**
- **Where to put JavaScript**
- **Variables**
- **Operators**
- **Looping**
- **Functions**

## 7. DATABASE DESIGN

- **Introduction**
- **Presentation Logic**
- **Business Logic**
- **Data Service Logic**
- **Architectural Design**
- **Conceptual Database Design**
- **Inputs**
- **Outputs**
- **Database Design**

## 8. ASP (3.0)

- **Introduction**
- **Advantage of using a Server-side Technology**
- **What is needed to run ASP?**
- **Installing Internet Information Services 5.0**

# CHAPTER # 1

# INTRODUCTION

# Introduction

The Internet has progressed at amazing speed in recent years. Once the realm of academics and defense agencies, Internet is rapidly becoming a mainstream media for communication between individuals, companies, and global dwellers.

As part of the Internet, World Wide Web is now the predominant force in growth. Its language is simple, its interface is attractive and friendly, and it is adaptable to a wide variety of uses.

There are now Web sites for selling products, selling ideas, maintaining appearances, informing public, continuing education and knowledge, and just waisting time. And in a growing trend, the Internet concept is being adapted to International communication by establishing Intranets inside companies.

Normally, man has to roam about here and there to get information to make investigation and to collect data, but now a days, it is preferable to sit in your home, just browse the web and get information.

We decided to take "Website Designing of Pakistan Telecommunication" as our project of PGD Computer sciences because we find not a single site, which can provide the basic information about telecommunication system of Pakistan.

We decided to take "Website Designing of Pakistan Telecommunication" as our project of PGD in Computer sciences because we find not a single site, which can provide the basic information about telecommunication system of Pakistan. Most of the telecommunication companies have made their own sites, which can only provide the information about the specific company and its services. In this way sooner or later either one gets frustrated or encounter shortage of time to see all around this city.

Keeping in view all the reasons, the need arises for a complete and comprehensive source of information about Telecommunication System of Pakistan. A source, which is self

sufficient to cater for the information services, easily accessible to all those who need to get information about Pakistan telecommunication.

This site will also help the people to know more about Pakistan Telecommunication. Different topics, which are included in this web site, will certainly fulfill the knowledge requirements of all the people.

## Objectives:

While designing this site we keep the following objectives in our mind.

- To make the site user friendly
- To make it easy and quick to access
- To provide all the relevant information at one single site
- To cover all the aspects of the need of this site
- To connect a database to the web pages

# CHAPTER # 2

# PROJECT OVERVIEW

# PTA

PTA stands for Pakistan Telecommunication Authority. It is regulatory body for telecom sector in Pakistan. Generally, the sector covers fixed-line telephony, mobile telephone, wireless communication, satellite consumer, Internet, cable TV, Autiotex, Paging service, Voice mail and Digital Radio Paging. Emerging technologies in telecom sector are expanding the list further.

The mission of PTA is to facilitate and ensure the availability of high quality, efficient, cost effective and competitive telecommunication services through out Pakistan and protect the interests and rights of consumers and licensees.

PTA came into being on 1st January 1996, under the Pakistan Telecommunication (Re-Organization) Act 1996.

This is a gross misconception. PTCL is a company providing telecommunication services. It is a licensee of PTA. PTA regulates the telecom sector as a whole and performs its functions independently.

## GROWTH OF TELECOMMUNICATION IN PAKISTAN

After the partition of Indo-Pak subcontinent in1947, the areas that became part of Pakistan were mostly neglected in respect of telecommunication services. The supporting organization and manufacturing of telecommunication equipment were almost non-existent in Pakistan.

### The Pakistan Posts and Telegraphs Department ( 1947 to 1962 )

In 1947, the Pakistan Posts and Telegraphs Department was attached with the Ministry of Communication. During the first fifteen years, a sound foundation was laid by creating supporting organizations like telephone stores, workshops, training centers, production and

repair of equipment, etc, necessary for running of PT&T Department. However, telecommunication network systems remained limited to major cities of the country. The Government of Pakistan started five-year plans to build a proper base for systematic development of the telecom sector.

## Pakistan Telegraph and Telephone Department (1962-1990 )

With expansion of the postal and telecommunication services, the government decided to bifurcate the PP&T Department into Pakistan Telegraph and Telephone Department and Pakistan Post Office Department. The process of bifurcation completed by July 1962.

Significant developments took place during the first forty years in terms of infra-structure development and transfer of technology from EMD to digital switching systems and increase in telephone lines from 12,000 in 1947 to 922,000 in 1990, besides establishment of manufacturing facilities of various types.

## Establishment of Pakistan Telecommunication Corporation (PTC)

In 1990, PT & T department was transformed into a corporation and titled as Pakistan Telecommunication Corporation. The objective of this initiative was to provide greater autonomy and flexibility to the organization in achieving its long-term objectives.

During the next five years, the telecommunications sector made tremendous progress in the provision of telecommunication services. It started manufacturing and production of telecommunication equipment/materials by using the latest technologies. During PTC period the number of lines increased to 21,26,054 in 1995, an increase of over 230 percent over 1990.

## Pakistan Telecommunication Company Limited (PTCL)

In December 1995, PTC was converted into a joint stock company under Pakistan Telecommunication (Reorganization) Ordinance, assets of the PTC were divided among Pakistan Telecommunication Company Limited (PTCL), Pakistan Telecommunication Authority (PTA), National Telecommunication Corporation (NTC) and frequency Allocation Board (FAB). PTCL inherited about 94.8 percent of PTC's assets, including

2.862 million access line installed (ALI) and 2.228 million subscribers (ALIS). Later, in October 1996, the parliament of Pakistan passed the Pakistan Telecommunication (Re-organization) Act, which established PTCL, PTA, NTC and FAB.

## Manufacturing Facilities

To meet with the requirements of telecommunication plants, switching equipments and other related material, the telecom department took essential measures. A brief description of major manufacturing facilities was as below:

The telecommunication manufacturing industry can be categorized as follows:

- Switching and transmission systems
- Optic fiber cables and systems
- Copper cables and networking accessories
- Telephone Sets and PABXs

## Telephone Industry of Pakistan (TIP)

To meet with the requirements of telecommunication plants, switching equipments, TIP located in Haripur Hazara, NWFP, was establishment in 1952. It was a private limited company owned jointly by PTCL and M/s siemens Germany with majority shares held by PTCL (71.65%).

The core activity of TIP was to manufacture and install digital telephone switching systems (proprietary technology of Siemens Germany). The factory also produced telephone sets, DP boxes, divisional cabinets and drop wire. Beside telecommunication equipment, TIP also manufactured a variety of other products like container shells, energy meters, fire alarm equipment etc.

The production of digital exchanges (EWSD) was introduced in 1990. Prior to that TIP was only manufacturing electro-mechanical telephone exchanges of German technology commonly known as EMD exchanges. TIP supplied and installed more than 700,000 EMD lines for PTCL from 1955 to 1990. In fact, the factory was the sole supplier of telephone exchanges to PTCL.

Since 1990, TIP had been producing more than 250,000 lines/ports of EWSD digital switches annually and was at present the largest manufacturer of digital telephone exchanges in the country. It had so far supplied more then 2.3 million local lines/trunk ports over the last ten years to PTCL and NTC for their digital exchanges all over the country.

### Zhongxing Telecom Pakistan (Pvt) Ltd (ZTE)

PTCL signed a contract with People's Republic of China's government owned telecom company, ZTE Corporation on October 17, 1998 for installing 266,000 digital telephone lines in Pakistan. An Agreement for technology transfer was also signed under which ZTE established a factory at Islamabad for production of digital switching systems. Under the agreement, ZTE's proprietary switching technology, ZXJ-10, would be transferred to PTCL in five years.

ZTE in also setting up a research and development center of international standard in Pakistan to support and continuously upgrade the ZXJ-10 switching system. The R&D center will also develop other indigenous high-tech telecommunication products.

ZTE products will be manufactured, both for the local market and for exports elsewhere. In its first two years of operation imported components will be assembled by local employees. By third year most of the circuit boards will be manufactured and by fifth year ZTE will manufacturing 100 percent of the ZXJ-10 switches in Pakistan. Exports from this factory will earn national prestige and useful foreign exchange for the country.

## PTA Structure:

The authority consists of three members one of whom is chairman. The Federal Government appoints the members. The Authority has six directorates, each headed by a Director General:

- Licensing and Regulations
- License Enforcement
- Policy and Research
- Finance and Tariff
- Law
- Technical

## PTA Functions:

- The Act specifies the Functions of the Authority.
- Section 4 (1) of the Act says:
- "The Authority shall...
- regulate the establishment, operation and maintenance of telecommunication systems and the provision of telecommunication services in Pakistan;
- receive and expeditiously dispose of applications for the use of radio-frequency spectrum;
- promote and protect the interests of users of telecommunication services in Pakistan;
- promote the availability of a wide range of high quality, efficient, cost effective and competitive telecommunication services throughout Pakistan;
- promote rapid modernization of telecommunication systems and telecommunication services;
- investigate and adjudicate on complaints and other claims made against licensees arising out of alleged contravention of the provisions of this Act, the rules made and licenses issued thereunder and take action accordingly;
- make recommendations to the Federal Government on policies with respect to international telecommunications, provision of support for participation in international meetings and agreements to be executed in relation to the routing of international traffic and accounting settlements;
- perform such other functions as the Federal Government may from time to time, assign to it."
- LICENSING
- You require a PTA license for the following services:
- Basic Telephony
- Electronic Information System
- Card Payphone
- Data Communication
- Store and Forward Fax
- Vehicle Tracking System
- Burglar Alarm System

- Cellular Mobile
- Trunk Radio
- Digital Radio Paging
- Audio Tex
- Voice Mail
- Satellite Communications

**VALUE ADDED SERVICES:**

Pakistan Telecommunication Authority has taken various steps for improving standard operating procedure for award of licenses for value added telecommunication services. During the year of report, PTA took steps to liberalize licensing for value added services like card payphone, internet, cable tv, data communication network services and audio tex etc. On the initiative of the Minister for Science & Technology, Prof Dr. Atta-ur-Rehman, PTA put in place adequate arrangements for processing licenses applications within seven days.

During the year 2000-2001, PTA issued 31 licenses for card payphone services, 26 for electronic information services, 9 for data communication network services, 5 for store and forward fax services, and one for vehicle tracking system.

| *S.No* | *Services* | *License Issued 2000-2001* |
|---|---|---|
| 1 | Card payphone Service | 31 |
| 2 | Electronic Information Services | 26 |
| 3 | Data Communication Network Services | 9 |
| 4 | Store & Forward Fax Services | 5 |
| 5 | Vehicle Tracking System | 1 |
| | Total | 72 |

# PTCL

Pakistan Telecommunication Corporation (PTC) was established in December 1990 to take over operations and functions from the Pakistan Telephone and Telegraph Department. Its operations were governed by the Pakistan Telecommunication Corporation Act 1991. At the same time the Government of Pakistan (GOP) began to introduce private participation in the sector and licenses were awarded for cellular, card-operated payphones, paging and more recently, for data communications services in the country.

In 1991, GOP first announced its intention to privatize PTC. In 1994, the Government of Pakistan decided to test the appetite of the domestic and international capital markets for PTCL. Consequently, in the third quarter of 1994, the Government of Pakistan issued six million 'Vouchers' exchangeable into 600 million shares (with a par value of Rs. 10 per share) of the future PTCL in two separate placements. These Vouchers were converted into shares of PTCL in mid 1996. Following such conversion the Government of Pakistan own (88%) and private investors own (12%) shares in PTCL.

In 1995, a Presidential Ordinance called the Pakistan Telecommunication (Reorganization) Ordinance was promulgated which established the basis for the monopoly of PTCL in the provision of basic telephony services in the country as well as the new independent regulatory regime for the development of the sector. In October 1996, the new Pakistan Telecommunication (Re-Organization) Act ("Act") was passed by the Parliament, which contains essentially the same provisions, as did the Ordinance, but on a permanent basis.

PTCL owns and operates the public switched telephony network in Pakistan and various other telecommunication services. In addition it is the sole provider of all Core Infrastructure services including international Data and Voice services on an exclusive basis till end December 2002. Over the last ten years a capacity expansion program succeeded in increasing telephone penetration, reducing pending applications and considerably improving productivity, digitalization and quality of the network.

The network expansion has been accompanied by steady and phenomenal expansion in national long distance (NWD) and international traffic. 1998 and 1999 saw the beginning

of PTCL's partnership with private sector enterprises for new projects such as pre-paid calling cards and wireless local loop. Currently PTCL is preparing for challenges of competition, which is likely to set early 2003. to be ready to face onslaught of competition PTCL has taken a number of proactive measures to outsource and contract out its rights under license granted by GOP/PTA.

## Performance of PTCL:

PTCL is a government owned joint stock company and provides besides basic telephony, besides several value-added services, such as e-mail, internet, digital cross-connect, ISDN, leased lines, call waiting/forwarding, speed dialing, toll-free numbers etc.

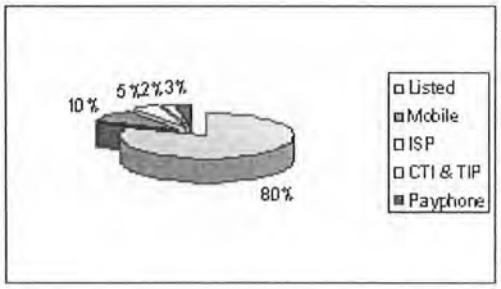
Pakistan inherited 12000 operational telephone lines at the time of independence in 1947.The then telegraph and telephone department (T&T) was converted to Pakistan Telecommunication Corporation in 1990 and then transformed into a public limited joint stock company in 1996. Like rest of the world, telecommunication sector in Pakistan is also going through a process of rapid change in convergence with focus on mobile, internet, cable TV and other value added services. PTCL is also moving to that direction in line with global trends. PTCL upgrade its network, introduced a range of new value-added service, developed a portfolio of IT, Internet and bandwidth related services during recent years.

## PTCL Transmission Network

PTCL expanded the outside plant (OSP) network for better utilization of spare switching capacity. During the year ending 30th June, 2001, a total of 265,026 digital lines were commissioned at 319 sites and surpassed the target for installation of new switching lines of 209,000 during the year. PTCL installed 67,386 lines in the rural

## Financial Size of the Telecom Industry of Pakistan (1999-00 & 2000-01)

| Operator | Rs in million 1999-00 | Rs in million 2000-01 | Percentage Growth |
|---|---|---|---|
| Listed | 144,154 | 153,372 | 6.4 |
| Mobile | 14,059 | 18,683 | 33.0 |
| ISP's | 7,968 | 8,796 | 10.4 |
| CTI, TIP | 3,172 | 3,635 | 14.6 |
| Payphone | 5,196 | 5,738 | 10.4 |
| Total | 174,549 | 190269 | 9.0 |




## Card Payphone

In terms of telecommunication-related developments, the global wave of technological advancements and innovations of the 1980's had been instrumental in the introduction of deregulation and privatization initiatives all around and card payphone was another area for the private enterprise. It served an important segment of the market, by providing access to occasional users and those who could not afford a private phone line. There was substantial demand, actual and potential in all telecommunication fields. Out of 3.25 million installed lines, 90 percent wee in urban areas, 51 percent of which were only in three major cities. Vast potential in smaller town and rural areas was still to be tapped. At present, the cellular mobile and card payphone services were also concentrated in three major cities of Karachi, Lahore and Islamabad/Rawalpindi. PTA had issued 68 licenses and 42 were operational by 30th June 2001.

## Main Payphone Operators with Number of Booths

| No | Name of Company | 1999-00 | 2000-01 | Growth (%) |
|---|---|---|---|---|
| 1 | Telecard | 6,325 | 7,454 | 17.84 |
| 2 | World Call | 6,726 | 12,667 | 88.32 |
| 3 | Call point | 8,526 | 12,343 | 44.76 |
| 4 | Telips | 2,150 | 2,834 | 31. |
| 5 | Telecom Foundation | 1,032 | 1,672 | |
| 6 | Global Telecom | 1,932 | 6,086 | |
| 7 | Voice Tel Tech | 844 | 1,792 | |
| 8 | Others | 175 | 12628 | |
| Total | | 27710 | 57476 | |

## RADIO-COMMUNICATIONS

Radio communications licensing covered amateur, aircrafts, ships and coastal stations services. The Frequency Allocation Board has exclusive powers to allocate radio frequency spectrum to all those applicants who intend to provide telecommunication services, operate radio and television or any other wireless service. The PTA receives applications for allocation and assignment of radio frequency spectrum and after initial examination refers them to the Frequency Allocation Board for allocation of frequency.

The total number of licenses issued for radio communication services increased from 2,056 in June, 2000 to 2,114 by June, 2001. The largest number of licenses issued during the year was 20 in VHF station category. The total number of licenses in this service was 691 at the end of June, 2001. It was followed by licenses for Inmarsats for hiking expeditions. PTA issued 13 licenses for this category during the year.

## RADIO COMMUNICATION LICENSES

| S.No. | Types of Radio Station/Service | Licenses as on June 2001 |
|---|---|---|
| 1 | Aircraft/Ship/ Coastal/Station | 68 |
| 2 | UHF Station | 212 |
| 3 | VHF Station | 691 |
| 4 | Broadcast Station | 381 |
| 5 | V-Sat Station | 6 |
| 6 | Spread Spectrum Radio Links | 30 |
| 7 | Amateur Station | 400 |
| 8 | Dealership | 18 |
| 9 | HF | 249 |
| 10 | Temporary uplink for special event | 2 |
| 11 | GMPCS | 1 |
| 12 | Satellite (for Data Communication) | 0 |
| 13 | Inmarsat for Expedition | 16 |
| 14 | Inmarsat Terminal | 40 |
| | Total | 2114 |

## CABLE TELEVISION

In pursuance of the government's policy to streamline cable TV network, PTA started issuing licenses to terrestrial cable TV operators from June, 2000.

PTA issued 822 CTV licenses by June 2001. Regional distribution of CTV licenses by office of issue was as under:

| | |
|---|---|
| Licenses issued at Islamabad/Rawalpindi/NWFP | 101 |
| Licenses issued at Lahore | 348 |
| Licenses issued at Karachi | 378 |
| Total | 822 |

## Cable TV Industry In Pakistan

Pakistan has been witnessing a growth in the demand for CTV services for quite some time. The driving force behind this demand has been the progressive conversion of satellite television signals beamed over this part of the world to digital and encrypted form. From only four channels in the early 1990s more than 100 TV channels had grown up over the last few years to serve the growing cable and home market. Some channels offered general entertainment whereas others offered special programmes through satellite.

For reasons of variety in programmes i.e. entertainment, news and sports, people in general resorted to install individual dish antennas. Rapidly falling TV prices, increasing relevance of TV in household entertainment, were also some of the drivers to enhance dish activity. Taking advantage of this situation, CTV operator sprung up in various localities to provide a range of programmes to residents.

## Profile of Telecom Industry In Pakistan

In, Pakistan, it is estimated that there are about 8.8 million households having TV sets, 4.2 million installed telephone lines and less than a million cellular connections. This given us a telephone penetration of around 2.4 per 100 peoples, one of the lowest in the world and certainly not the level of infrastructure expected for fast developing economies. There are about 1.3 million people with Internet access and around 2 million cable TV subscribers. There are more than 800 licensed cable TV network operators throughout Pakistan, a large number of which are small operators with less than 500 subscribers each. An undetermined number of small cable operators in isolated locations are also running the business without having a license and caring a legal action, which may be initiated at any time.

With the rise in Internet access, an increasing number of CTV companies around the world have started adjusting their networks and business plans to take advantage of cable modem technology. Traditional Teleceos are moving to acquire CTV operations or deploy their own cable TV systems. The ultimate goal is to integrate Internet access, voice service and video into a package that could be delivered through the same pipe to individual homes or business. Pakistan is probably one of the countries in the region where Government had

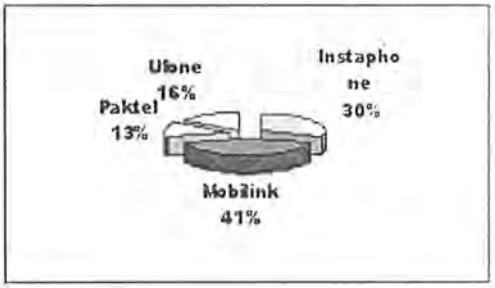
realize the potential of cable and allowed Internet over cable as the first transitional step towards a regime of more value added services.

## Mobile Services

Four companies are providing cellular mobile services in Pakistan i.e. Paktel, Instaphone, Mobilink and Ufone. Mobilink and Ufone are using digital technology (GSM) for its cellular service. Lately Instaphone introduced digital version of the analogue technology i.e. D-AMPS, (Digital Analogue Mobile Phone System).

In November 2000, Pakistan Telecommunication Authority issued a determination on 'Calling Party Pays' tariff regime. Soon after all the companies adopted this regime. The effect of this regime was that incoming calls on mobile phones became free. Earlier to that, the mobile users had to pay for incoming calls as well. This had a deterring effect on the user. He would avoid telling his mobile number to others and would also be tempted to keep the phone short. With the introduction of CPP, the mobile users were encouraged to use the mobile phones.

### Market Share Of Cellular Operators In Pakistan 2000-01




## Internet Services

Realizing the need of the time, Government of Pakistan initiated the process of deregulation telecom sector. This process took a jump start with the efforts of the Minister of Science & Technology Prof. Dr. Atta-ur-Rehman. The Internet was available in over 530

cities of Pakistan by end of 2000. Bandwidth rates were reduced drastically. Software industry was given incentives. Human resources development in the field of information technology was being addressed on priority rather emergency basis. All this was necessary and important particularly introduction and strategically planned spread of Internet.

By June 2001, Pakistan Telecommunication Authority had issued 146 licenses to provide Internet Services out of which 60 had started operation with estimated number of 500,000 users. The size of the bandwidth currently increased from 50 Mbps to 100 Mbps. The total number of lines acquired by ISPs was about 21,000 as PRIs, and analogue lines. The amount of capital invested by ISPs was about 21,000 as PRIs, and analogue lines. The amount of capital invested by ISPs was over Rs 2 billion so far.

In order to promote Internet in the country, PTA adopted an open license policy and introduced a state of competition in the market resulting a lower price for its usage and improved quality. However, another step to promote the industry is would be to educate people about the use and possible benefits of Internet through media.

## Calling Cards

In Pakistan, there are four companies providing the facility of calling cards.

- WorldCall Calling Cards
- CallPoint Calling Cards
- Callmate Calling Cards
- PTCL Calling Cards

PTCL Calling Cards the pre-paid calling card that gives you nationwide access alongwith International facility.

Dancom started its prepaid calling card services (PCCS) in June 2002 under existing brand name CALL POINTs Calling Card. The calling card service is essentially under an O&M contract with sole PSTN operator Pakistan Telecommunication Company Limited (PTCL).

WorldCALL undertook another pioneering effort when it became the first company to introduceinternational prepaid calling card service in Pakistan.

WPL launched its operations under the service brand name Hello ® in March 1999 and in a short span of time gained significant market share. The service was started in association with Unitel Communications, UK. Unitel is the turnkey technology supplier and also initially responsible for maintenance of the IN Platforms.

Three IN platforms have been installed in Karachi, Lahore and Islamabad and the combined capacity now extends to 2000 ports. Domestic long distance dialing service feature was added on the same card in year 2000. Out of four service providers, including the state run monopoly Pakistan Telecommunications Company Limited (PTCL), WorldCALL holds 30% market share.

## Important Telephone Numbers

In this web site, different telephone numbers of different cities of Pakistan are provided which is in a table made in access and is connected to the web pages by using ASP at the front end.

## Codes

In this site, telephone codes of all the cities of Pakistan and different countries of the world are provided. The tables are generated in access and is connected to the web pages by using ASP at the front end.

**"The site map as well as the user guide is provided at the end of the project."**

# CHAPTER # 3

# SOFTWARE USED

# SOFTWARES USED

It was decided to put all the collected information in the form of web pages, so that all can easy access it. To design web pages, the best choice that we can chose was to use HTML language and to give them an attractive, sober and scholarly look.

In order to achieve the above objectives, we use the following tools.

- **HTML:**

  The language used to develop web pages is called Hyper Text Markup Language (HTML).

- **Javascripts:**

  Javascript is an easy-to-use programming language that can be embedded in the header of your web pages to enhance the dynamics and interactive features of web page

- **Swish:**

  SWiSH is quick, easy and affordable way to create Flash animations for the web site.

- **Microsoft Access:**

  MS Access is used to create Database, which is later connected with web pages.

- **ASP:**

  Active Server Pages (ASP) is a great tool for creating dynamic web pages. This tool is used for connecting the Database created in MS Access to the web.

- **Word Processor Used:**

  - Notepad is used to write HTML programs.
  - MS Word 2000 is used to write text for the web site.

- **Browsers Used:**

  - Microsoft Internet Explorer 6.

- **Hardware Environment:**

  To get Optimum performance from the software, the system should be IBM compatible Pentium II or higher, RAM 32 MB or more.

- **Operating System:**

  Windows 2000 NT.

# CHAPTER # 4

# INTERNET BASICS

### What's in a Web browser?

A Web browser contains the basic software you need in order to find, retrieve, view, and send information over the Internet. This includes software that lets you:

- Send and receive electronic-mail (or e-mail) messages worldwide nearly instantaneously.
- Read messages from newsgroups (or forums) about thousands of topics in which users share information and opinions.
- Browse the World Wide Web (or Web) where you can find a rich variety of text, graphics, and interactive information.

Browsers such as Microsoft Internet Explorer 6 include additional Internet-related software. For example, with Internet Explorer 6, you also get:

- Outlook Express messaging and collaboration client
- Microsoft Windows Media Player
- NetMeeting conferencing software
- ActiveX controls
- Chat
- DirectX application programming interface
- Subscriptions for automatic Web page updates
- Dynamic hypertext markup language (HTML)

## What is a URL?

A URL (or uniform resource locator) is the address of an Internet page on a Web site. Usually it consists of four parts: protocol, server (or domain), path, and filename. Often, when you go to the very first page of a Web site, called the home page, there's no path or filename. Here's an example:

**http://www.microsoft.com/windows/default.asp**

- http is the protocol
- www.microsoft.com is the server
- windows/ is the path
- default.asp is the filename of the page on the site

## Browser Tips:

Use the following tips to take advantage of all the great features of Microsoft Internet Explorer 6.

### Spot the links

You can tell whether an item on a page is a link by moving the mouse pointer over the item. If the pointer changes to a hand, then the item is a link. A link can be a picture, a three-dimensional image, or colored text. Click any link on a Web page to go to another page within that site or another site.

### Display all Web pages faster

To display Web pages faster:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. Click the *Advanced* tab.
3. In the Multimedia area, clear one or more of the *Show pictures*, *Play animations*, *Play videos*, or *Play Sounds* check boxes.
4. If the *Show Pictures* or *Play Videos* check box is cleared, you can still display an individual picture or animation on a Web page by right-clicking its icon and then clicking *Show Picture*.
5. If the pictures on the current page are still visible after you clear the *Show pictures* check box, you can hide them by clicking the *View menu* and then clicking *Refresh*.

## Display previously viewed pages faster

To display previously viewed pages faster:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. On the *General* tab, click *Settings*.
3. To create more space to store pages temporarily, move the slider to the right.
4. To prevent Internet Explorer from updating pages in the Temporary Internet Files folder, click *Never*.

## Change how page colors are displayed

To change how page colors are displayed:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. Change the settings as needed.

## Display text in a different font

To display text in a different font:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. On the *General tab*, click *Fonts*.
3. In the Proportional and Fixed-width font lists, click the fonts you want.

## Specify which font and color setting to always use

To specify which font and color settings to always use:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. On the *General* tab, click *Accessibility*.
3. Change the settings as needed.

## Display text larger or smaller.

On the *View* menu, point to *Text Size*, and then click the size you want.

## View Web pages in a different language

Some Web sites offer their content in several languages. You can add languages to your list of languages in Internet Explorer so that you can view these sites in your preferred language. To view Web pages written in a different language:

1. On the *Tools* menu in the browser, click *Internet Options*.
2. On the *General* tab, click *Languages*.
3. Click *Add*.
4. Select the language you want to add.
    - If you speak several languages, you can arrange them in order of priority. If a Web site offers multiple languages, it will supply content in the language with the highest priority.
    - Adding languages does not guarantee that you have a font that can display Web pages in your preferred languages. You may need to download a multilanguage support pack to display pages in these languages..

## Add a page to your Favorites:

To add a page to your collection of favorite pages:

1. Go to the page that you want to add to your collection of favorite pages.
2. On the *Favorites* menu, click *Add to Favorites*.
3. Type a new name for the page if you want to.
    - To open one of your favorite pages, click the *Favorites* button on the toolbar and then click the page you want to open.
    - To keep track of your favorite pages, you can organize them into folders. Click the *Create In* button in the *Add to Favorites* dialog box.

## Add Microsoft Product Insider to your Favorites

To make sure you always have access to the latest Internet news, software updates, and tips and tricks for using Internet Explorer and other Microsoft products, why not add the Microsoft Product Insider site to your Favorites list now? To add Product Insider to your Favorites list, follow these steps:

1. On the *File* menu of your Internet Explorer toolbar, point to *New* and click *Window* so you don't lose your place in this guide.
2. On the *File* menu in the new window, click *Open*, and then type **http://www.microsoft.com/insider/** in the address box.
3. Click *OK*.
4. When the Microsoft Product Insider home page has finished loading, on the *Favorites* menu, click *Add* to *Favorites*, and click *OK*.
5. Close the new window.

## Organize your Favorites into folders

To organize your favorite pages into folders:

1. On the *Favorites* menu, click *Organize Favorites*.
2. Click *Create Folder*, type a name for the folder, and then press ENTER.
3. Drag the shortcuts in the list to the appropriate folders.
    - You might want to organize your pages by topic. For example, you could create a folder named Art for storing information about art exhibits and reviews.
    - If the number of shortcuts or folders makes dragging impractical, you can use the Move button instead.

## Find Favorites more quickly in an overloaded Favorites menu

To scan a large Favorites menu more quickly:

1. On the Windows *Start menu*, point to *Find* and click *Files or Folders*. Windows will display a *Find: All Files* dialog box.
2. In the *Look in* dropdown box, type *c:\windows\favorites*, or browse to this directory.
3. In the *Named* dropdown box, type the filename you are looking for and click the *Find Now* button. Windows will display all the Favorites that match your query and list information about each.
4. If there are multiple results, you can click the column information title and sort the results by name, date, and so on.

## Edit Favorites

You can do a lot more with your Favorites folder list in Internet Explorer 6 than you can with other browsers. Remember these tips:

- You can drag a Favorite or folder to different areas to reorganize.
- By going to the Start menu, pointing to Find, and clicking on Files or Folders, you can select a Favorite or folder from the list box and drag it to your desktop.
- Right-click a Favorite or folder to display a pop-up menu that lets you perform functions like Edit (in the Microsoft FrontPage® Web site creation and management tool), Subscribe, Copy, and Delete.
- You can click the Send To option to send the Favorite to a floppy disk, create a shortcut on the desktop, or attach the shortcut to an e-mail message.

## Change your home page

To change your home page:

1. Go to the page that you want to appear when you first start Internet Explorer.
2. On the *Tools* menu, click *Internet Options*.
3. Click the *General* tab.
4. In the Home Page area, click *Use Current* (to restore your original home page, click *Use Default*).

## Save text and graphics from the Web

When you see text or graphics on a Web page that you like or want to refer to later, you can save them on your computer's hard disk. Later, you can open the saved file and review it offline.

To save a text or source file:

1. On the toolbar, click *File*, and then click *Save As*.
2. Click *Save* to save the file.

To save a graphic:

1. Right-click the graphic.
2. On the shortcut menu that appears, click *Save Picture As*.
3. Browse to the folder where you would like to save the file.
4. Click *Save* to save the file.

To open a saved file, double-click it from the folder where you've saved it. Internet Explorer will start automatically, and your saved file will appear in the browser window.

## Add a page to your Links bar

To add a page to your Links bar:

- Drag the icon for the page from your *Address* bar to your *Links* bar.
- Drag a link from a Web page to your *Links* bar.
- Drag a link to the *Links* folder in your *Favorites* list. You can drag it directly to the *Favorites* menu and then into the *Links* folder, or you can drag it to the *Links* folder when displaying your Favorites in the Explorer bar.

You also can organize your links by dragging them to a different location on the Links bar.

## Create a desktop shortcut to the current page

Right-click the page, and then click *Create Shortcut*. If the Internet Explorer window is not maximized, you also can create a shortcut by dragging a link from the Internet Explorer window to the location you want, such as your desktop or a folder.

## Return to a Web page you've already seen

There are several ways to return to a previously viewed Web page:

- To return to the last page you viewed, you can click the *Back* button on the toolbar or press the *BACKSPACE* key.
- To see a list of the last few pages you visited, click the small down arrow beside the *Back* or *Forward* button. Then click the page you want.
- If you want to view one of the last five pages you visited in this session, click the *File* menu and click the page that you want to go to. This list is started fresh every time you start Internet Explorer.
- To view more pages, including pages you visited in previous sessions, click the *History* button on the toolbar and then click the appropriate folder.

## Change the appearance of the toolbar

To change the appearance of the toolbar:

- You can move or resize the *Address* bar and *Links* bar by dragging them up, down, left, or right. You can even move them into the menu bar. If they won't move, right-click on the bar and uncheck *Lock the Toolbars*, then left-click on the word *Address* or *Links* and drag the bar where you want it to go.
- To make more room on your screen, you can hide toolbar button labels. Just right-click the toolbar, click *Customize*, and select *No text labels* from the *Text options* dropdown list.
- You can hide the *Address* bar or *Links* section of the toolbar by right-clicking the toolbar and clearing the check mark for each item you want to hide.

- You can add items to the *Links* bar by dragging the icon from the *Address* bar or dragging a link from a page.
- You can rearrange items on the *Links* bar by dragging them to a new location on the bar.
- You can use smaller Microsoft Office-style toolbar buttons. On the *View* menu in a browser window, select *Toolbars*, then select *Customize*. Under *Icon options*, choose the size you want.

## Use pop-up menus for quick access

Internet Explorer 6 features pop-up shortcuts to functions like Save As, Open, and Copy. To access a pop-up menu:

1. Place the mouse cursor on a hypertext link, graphic image, or Web page, and right-click.
2. Internet Explorer will display one of three pop-up menus you can use to quickly perform the desired function. Pop-up menu functions include:

Add to Favorites
AltaVista Home
AV Translate
Back
Copy Background
Copy Shortcut
Create Shortcut
E-mail Picture
Encoding
Forward
Go to My Pictures
Open Link
Open Link in New Window
Print Picture
Print Target
Properties

Refresh

Save Background

Save Picture As

Save Target As

Select All

Set as Background

Set as Desktop Item

Show Picture

View Source

## The Cache and Toolbar

### Cache

When you explore the World Wide Web, your browser keeps track of the pages you've visited and saves them on your hard disk so they'll load faster when you return to them. This saves you time and money because you can view the saved pages without being connected to the Internet. The saved files, your "Temporary Internet Files", are stored in your disk cache.

**To empty your disk cache**

When you browse, your disk cache can fill up with files you no longer need. Here's how to empty your Internet Explorer disk cache.

For Internet Explorer 6:

1. On the *Tools* menu of your Internet Explorer toolbar, click *Internet Options*.
2. Click the *General* tab.
3. In the Temporary Internet Files area, click *Delete Files*, and then click *OK*.
4. Click *OK* to close *Internet Options*.

**To change the size of your Internet Explorer disk cache**

You can change the amount of hard-disk space reserved for your disk cache. A larger disk cache may display previously visited pages faster, but it will decrease the amount of hard-disk space available for other files. Here's how to set the size of your disk cache.

For Internet Explorer 6:

1. On the *Tools* menu of your Internet Explorer toolbar, click *Internet Options*.

2. Click the *General* tab.
3. In the Temporary Internet Files section, click *Settings*.
4. In the Temporary Internet Files Folder section, drag the arrow on the *Amount of Disk Space to Use* slider to the percentage of disk space you want designated for your disk cache, and click *OK*.
5. Click *OK* to close Internet Options.

**Toolbar**

The Microsoft Internet Explorer toolbar consists of buttons that are shortcuts for menu commands. They make browsing faster and easier.

**Back.** Lets you return to pages you've viewed, beginning with the most recent. With Microsoft Internet Explorer 6, right-click the *Back* button and select from a list of recently visited sites.

**Forward.** Lets you move forward through pages you've viewed using the *Back* button. With Internet Explorer 6, right-click the *Forward* button and select from a list of recently visited sites.

**Stop.** Halts the process of downloading a Web page. Click this if you want to stop downloading a page for any reason—for example, if you're having trouble downloading it or if you don't want to wait for it to download. Then try downloading it again or browse elsewhere.

**Refresh.** Updates any Web page stored in your disk cache with the latest content. When you return to a page that you've visited, your browser displays the file stored in your disk cache, rather than the current page on the World Wide Web. This saves download time.

**Home.** Returns you to your home page. You can designate any Web page as your home page.

**Search.** Displays a choice of popular Internet search engines in the left pane. Your search results appear in the left pane, too. When you click a link, the page appears in the right pane, so you don't lose sight of your search results.

**Favorites.** Displays a list of the sites—and, with Internet Explorer 6, the folders, files, and servers—that you've saved as Favorites. Click any item in the list to jump to it.

**Media.** Displays a list of audio and video media options.

**History.** Shows a list of Web sites you've visited.

**Mail.** Connects you to the Microsoft Outlook Express messaging and collaboration client so you can read e-mail and newsgroup messages.

**Print.** Prints the page you're viewing. This is one way to save information from the Internet so that you don't have to reconnect to view it again. You can even print the URL associated with each hyperlink, making it easy to navigate to the site later.

**Edit.** Opens a file in the Microsoft Word word processor that contains the HTML code for the page you're viewing so you can see and even edit it.

**Messenger.** Opens Windows Messenger.

# CHAPTER # 5

# HYPER TEXT MARKUP LANGUAGE (HTML)

# HYPER TEXT MARKUP LANGUAGE (HTML)

## Introduction To HTML:

Tim Berners-Lee, founder of the World Wide Web, offers this definition; "Hypertext Markup Language (HTML) is a simple data format used to create hypertext documents that are portable from one plateform to another". This definition was taken from a November 1995 memorandum from Berners-Lee regarding HTML 2.0.

Portability and simplicity are the two hallmarks of HTML. We can create HTML files on any computer that has Web browsing software. If we have web browsing software and a connection to the Internet, we can also view any HTML files available on the World Wide Web. We can move HTML files from one type of computer to another with no loss of meaning. HTML is easy language to use.

HTML was developed specifically for use on the Web. We are not restricted to using HTML on the Web, however; HTML has a wide range of applications outside that Venue. Companies use HTML as a document format for internal use, and individuals use HTML to facilitate the exchange of documents. More than 30 million HTML documents are available on the Web, and HTML and the web have become almost synonymous in the minds of the general public. HTML and the Web are not same, though. HTML is the standard method we can use to put information into a universally readable format on the Web.

### History of Hypertext:

The obvious first question when defining HTML is "what is hypertext?" start by looking at the word it self. Hyper has a connection of extra or beyond when used in words such as hyperactive and hypersensitive. Text refers to a written block of language, whether it is prose, poetry, a glossary, or any other form of written information. Ted Nelson coined the term hypertext in 1965 to refer to textual information that is not limited by the bounds of

the text itself. Hypertext can be connected to the other texts, and it can be navigated by a system of links in non-sequential order.

Hypertext goes beyond plain text. It is not designed to be read in a linear fashion, form beginning to end, but selectively, as readers require certain information. Hypertext is easier to demonstrate than to define. We are probably familiar with a basic hypertext system the online help that most software programs today features. We can go directly to the subject on which we want information, and most online help entries also include links to related topics.

### *The Best Online Help Programs:*

If we can't find what we are looking of with one method of searching, another method usually works. When using online help, if we already know to implement a certain features, it is not compulsory we have to page through an explanation of that features for the fifteenth time. Another example of a hypertext system is an encyclopedia on CD-ROM print encyclopedia has always encompassed a rudimentary sort of hypertext. Readers choose specific articles; they rarely (if ever) read and encyclopedia from beginning to end.

Many articles include suggestions of other article, to read. Choosing the text article probably means that we want to return our current volume to the shelf, pick up another volume. Search for the desired article read it, see the articles suggested next, and then perhaps repeat the process again and again. Encyclopedias on CD-ROM streamline this process significantly. We can jump in then investigate the ancient Phoenicians.

Reading in the conventional manner is still possible, but hypertext reading is encouraged by the medium itself. The antithesis of hypertext is an ancient scroll, which requires the reader to unfurl the text in its entirety from the beginning to the desired point in order to locate a specific passage. In contrast, a printed book that includes a table of contents and an index encourage a certain degree of nonlinear exploration. The reader can look for specific information and jump between topics. The use of hypertext is a natural response to the information explosion of the 20$^{th}$ century.

In the 16th century, some people might have a considerable percentage of all the books in existence and had at least a general knowledge of most academic subjects. Plodding through several volumes to find a specific fact might have been acceptable pastime for person of leisure. No one today, though, could possibly be expected to have read all the books, journals, newspapers, and other information sources currently in existence in order to keep up to date, even in a specialized field.

As Vannevar Bush, often consideration the father of hypertext, wrote in a 1945 article, "The investigator is staggered by the findings and conclusions of thousands of other workers conclusions which he cannot find time to grasp, much less to remember, as they appear." (Bush's article was first published in the July 1945 issue of The Atlantic Monthly, and it was republished, with permission, at the following address: http://www.isg.sfu.ca/~duchier/misc/vbush/.)

Bush's reflection on the increasing vastness of human knowledge is even truer today. But the development of workable hypertext systems that categorize huge area of knowledge is making the task of finding specific fasts and information much easier than it once was. Properly designed hypertext facilities indexing and categorization. Researchers in any subject area understand how helpful a good index can be. Imagine the indexes of tens of thousand of books and articles being made available in a huge database, and all this information being accessible with a very user-friendly search mechanism.

Then imagine being able to search by any keyword in any text or group of texts. The Web, at its best, includes such as indexed system, and your documents can be part of it. Search engine such as Alta vista, at http://a;tavista.digital.com now we perform keyword searches of all documents available on the web (or at least the majority of these documents), which have been made available to Alta Vita's indexing robots. Hypertext at its best can allow us to pinpoint the exact information required, no matter what the rest of the document or documents look like. However, having our documents in hypertext format is no substitute for basic writing skills. In fact, clear writing may be more important in respect to hypertext document than it is for regular documents. With hypertext, we can't rely on our user knowing or recognizing the underlying context we've built up elsewhere, as we can with a regular document. To maximize the effectiveness of our hypertext

documents. concept of clarity at the forefront when we create our documents. we're less likely to experience problems later.

In the context of modern computers, the term hypertext is something of a misnomer. Because graphics and other media, not just text. frequently play a major today's hypertext system. The term hypermedia, also coined by Ted Nelson, more accurately describes the type of structured documents that let your navigate seamlessly among text. graphics, sounds, and video. For example, if we perform a search for Mozart on a CD-ROM encyclopedia, we'll probably be able to hear an excerpt of one or more of the composer's works, in addition to reading his biography and linking to related articles. Although the web was largely (but not entirely) text based at the time of its inception in 1990, other forms of information are playing an increasingly important rote. The term hypertext Markup language is probably here to stay, but keep in mind that HTML is a document format.

## History of HTML:

Today, the advantages of hypertext are apparent to many people. However, hypertext was a little used concept before the widespread use of personal computers at least in the sense of the powerful implements you see to day. The hypertext system proposed in 1945 by Vanneva Bush. The memex involved a cumbersome apparatus that would have used, among other things, a microfilm machine and various photography techniques.

Before CD-Rom drives were first installed in PCs in the late 1980, relatively few cheap and easily searchable storage mechanisms were available for vast quantities of data. The storage capacity of hard drives was minimal compared to today's standard drives, which can hold gigabytes of data. Microfilm can hold a great deal of information, but performing searches in microfilm is time consuming and tedious.

In that proposal, Berners Lee describes the difficulties of managing information flow efficiently among the more than 2000 staff members of CERN, in a workplace ranging over several geographic locations and involving high employee turnover. The use

of several different computer systems complicated file and information sharing, because the formats often were incompatible.

Berners Lee proposed a system that would allow information to be shared across a variety of plate forms. The information was to be accessible regardless of the type of computer the user had. Furthermore, in order to aid users who needed specific and precise information, all information was to be presented in hypertext format and cross-referenced to other documents. The genius of Berners Lee was to propose the marriage of cross plateform compatibility and information presented on in hypertext format. Even more significantly, he proposed a system that would actually make such information interchange feasible. The 1989 paper was the genesis of the web (the term World Wide Web was developed in 1990). The mechanism for putting information into hypertext format developed into HTML, various computer-networking protocols have been developed to facilitate communication between entirely different systems.

## What is HTML?

HTML is the format in which we can put our information for display on the World Wide Web. Every HTML file is an ASCII text file (also called a plain text file), even if it refers to other forms of media, such as sound or graphic. We can open any HTML file in a text editor or even a word processor. To get an HTML file for viewing, connect to Internet provider, and then visit the Web site.

Follow the instructions in browser to save the HTML file, which combines the actual text we can see on the site with a bunch of codes. In Netscape, select file! Save As. Once we have done this, we can see the actual coded HTML file. Open the newly obtained HTML file in a text editor or word processor, we can see it. In the text file, we can see that all the HTML instructions are inside angled brackets like this: <TAG>.

Our browser interprets these instructions to mean that the textual information should be displayed in a certain manner. If other media (Such as sound, graphics, animation, and video) are referred to in the HTML file, our browser will display them (if it has that capability). If our system can't interpret capability, we won't hear any sound files.

If our browser can't display images, we won't see any images. Ideally, we will design our Web pages so that anyone with any browser can make sense of them, even if they can't access all the information. HTML is designed in away that accommodates these differing capabilities. If pages are coded properly, people who can't see images or video will see descriptive text instead.

## HTML is not a Page Description Language:

Before we discussion further detail what HTML is and how it works, it would be useful to consider what HTMl is no. HTML is very powerful, but there are some things it can't do. Many inexperienced Web author think of HTML as a formatting language that let we place images and text on the screen as precisely as we can create a document on paper.

But HTML isn't page description language. A page description language, such as those used in page market and similar programs, describe to the tenth of an inch where margins should go, the exact size style of font that should be used, the precise color of the text, and so on. HTML can't make those kinds of formatting decisions.

## HTML TAGS:

Tags are instructions that are embedded directly into the text of the document. An HTML tag is a signal to a browser that it should do something other than just throw text up on the screen. By convention all HTML tags begin with an open angle bracket (<) and end with a close angle bracket (>).

HTML Tags are discussed as follows.

### ➢ HTML:

The first and last tags in a document should always be the HTML tags. These are the tags that tell a Web browser where the HTML in your document begins and ends.

The absolute most basic of all possible Web documents is:

<HTML>

<HTML>

That's it. If you were to load such a page into a Web browser, it wouldn't do anything except give you a blank screen, but it is technically a valid Web page. Obviously, you'll want more than that.

### ➢ HEAD:

The HEAD tags contain all of the document's header information. When I say "header," I don't mean what appears at the top of the browser window, but things like the document title and so on. Speaking of which...

### ➢ TITLE:

This container is placed within the HEAD structure. Between the TITLE tags, you should have the title of your document. This will appear at the top of the browser's title bar, and also appears in the history list. Finally, the contents of the TITLE container go into your bookmark file, if you create a bookmark to a page.

What you type should probably be something, which indicates the document's contents, but it doesn't have to be. The length of the title is pretty much unlimited, but don't go overboard. Users will either sneer at or be confused by exceedingly long titles.

If you don't type anything between the TITLE tags, or don't include the TITLE tags at all -- remember the blank document in the HTML section earlier? -- Then the browser will typically use the actual file name for the title. Therefore, a document titled "TCh4ex4.html" will have that name appear in the history list. Again, you can choose to do this, but it will likely generate either confusion or contempt.

You should only have **one** TITLE container per document. At one point, it was possible to create "dancing titles" by including multiple TITLE tags. Not only is this a savage abuse of HTML, but also the effect can only be seen in certain versions of certain browsers. Therefore, it should be avoided at all costs.

## ➢ BODY:

BODY comes after the HEAD structure. Between the BODY tags, you find all of the stuff that gets displayed in the browser window. All of the text, the graphics, and links, and so on -- these things occur between the BODY tags. We'll get to what happens there starting with the next chapter.

So, putting everything we've covered thus far into one file, we have:

```
<HTML>
<HEAD>
 <TITLE>Document Title</TITLE>
</HEAD>

<BODY>

</BODY>
</HTML>
```

This time, the result would be a document with a completely blank browser window, but at least the words "Document Title" would appear in the browser's history list.

Let's look at the above block of HTML again, but this time with container lines sketched in. Note that the TITLE tags and text have been rearranged to make it easier to draw in the container lines. The rearrangement of the text does not in any way change the resulting Web page's appearance.

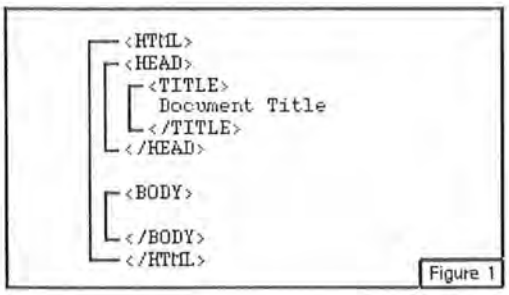



Figure 1

## ➢ COMMENT TAGS:

If you want to leave yourself notes in an HTML document, but don't want those notes to show up in the browser window, you need to use the comment tag. To do that, you would do the following:

```
<!-- Hi, I'm a comment. -->
```

Your note would go where the text Hi, I'm a comment appears. Yes, you do need an exclamation point after the opening bracket, but **not** before the closing bracket. That's the way the standard is written. I have no idea why. Also, there is no end tag; that is, a tag like </!-- text --> does **not** exist. The comment tag is **not** a container. This is our first example of an empty tag.

You can put comments pretty much anywhere, but you have to be aware of one important thing: you shouldn't put any HTML markup within a comment tag. Theoretically, you should be able to, but most browsers handle this less than gracefully (i.e., they either mess up or crash).

What if you get the tag wrong, like forgetting to include the exclamation point? In that case, the text you did type in would be displayed.

## ➢ HEADINGS:

The heading structures are most commonly used to set apart document or section titles. For example, the word "Headings" at the beginning of this section is a heading. So is this document's title (it's at the top of the page, in case you somehow missed it).

Remember that these heading structures go into the body of the document. The headings being discussed here have nothing to do with the HEAD structure from the previous chapter.

There are six levels of headings, from Heading 1 through Heading 6. Heading 1 (H1) is "most important" and Heading 6 (H6) is "least important." By default, browsers will display the six heading levels in the same font, with the point size decreasing as the

importance of the heading decreases. Here are all six HTML pairs, in descending order of importance:

```
<H1>Heading 1</H1>
<H2>Heading 2</H2>
<H3>Heading 3</H3>
<H4>Heading 4</H4>
<H5>Heading 5</H5>
<H6>Heading 6</H6>
```

These six lines, when placed into an HTML document, will simply display the six levels of headings.

Since, as we have discussed, *whitespace doesn't matter*, you might think that the above block of HTML would just string the content into one line of text. However, because headings are meant for section titles and the like, they are defined as existing on a line by themselves. A heading always begins at the margin of a line and always forces a line break at the end of the heading. In other words, you cannot have two heading levels on the same line.

This also means that you cannot highlight text in the middle of a paragraph by marking it as a heading. If you try this paragraph will be split in two, with the heading text on its own line between the two pieces.

If the browser which is set close to its default settings, you'll notice that the text for the last two headings gets pretty small. This leads to some page designers using H6 for the fine print at the bottom of pages. This is a mistake, not to mention an abuse of the heading structure. As you no doubt know, many browsers allow the user to set the size of each element, including the headings. If a user sets H6 to a size of 18 point, the fine print won't be so fine any more! Remember: *you cannot guarantee that your document will appear to other people exactly as it does to you.*

## ➢ PARAGRAPHS:

As you might suspect, paragraphs are quite common in Web pages. They are one of the most basic structures in HTML. If you regard a document as a collection of structures and sub-structures, you may come up with something like:

> The overall structure is a page. The page is composed of a number of sections, each of which is composed of one or more paragraphs. Each paragraph is composed of words, and each word of letters.

Admittedly, this is a simplified way of looking at text, but it will do for our purposes. The furthest HTML goes down this progression is to the paragraph level.

The beginning of a paragraph is marked by <P>, and the end by </P>.

At this point, you should once again be saying to yourself: *whitespace doesn't matter*. You could put each word on its own line, and the paragraph would look completely normal. In fact, no matter how much whitespace you put between words, whether returns or spacebar hits, the words will be separated by one space in a Web browser.

## ➢ LINE BREAK:

So what if you want to end a line after a certain word, but don't want to start a new paragraph? Well, what you need is a line break, which is invoked by using the <BR> tag. This forces a line break wherever you place it in the content (that is, whatever is after the <BR> tag will start from the left margin of the next line on the screen.)

And no, there is no </BR> tag. The line break tag is -- that's right! -- an empty tag. And when you think about it, this makes sense. The concept of a line break beginning and ending doesn't really work, since a line break is a one-shot occurrence.

Try it yourself, or look at an example.

## ➢ BLOCKQUOTE:

Blockquotes are handy for those long pieces of text, which are quoted material and therefore need to be set apart and indented. That is exactly what blockquote does. For example:

> This section of text is surrounded by the blockquote tags. A blockquote can exist inside of a paragraph, and always lives on its own line (which is to say, there is an implied line break before and after the blockquote, just as with headings or paragraphs themselves).

Blockquotes are set up as follows:

<blockquote> ...*text*... </blockquote>

Just like most other things in HTML, it's a container.

## ➢ LISTS:

There are three main types of lists. I've included the heading here because lists are basic text structures -- they just need a lot more explanation. That's what you'll find in the next section.

While simple in concept, lists can be very powerful in execution. There are three types of lists: unordered lists, ordered lists, and definition lists. The first two are very similar in structure, while definition lists have a unique setup.

## ➢ UNORDERED LISTS:

The term "unordered list" may be a bit unfamiliar to you, but odds are you've heard of the "bullet list." That's exactly what an unordered list is -- a list of items, each one preceded by a "bullet" (a distinctive character; typically, a small black circle).

The list begins and ends with the tags <UL> and </UL> respectively. Each item in the list is marked using the <LI> tag, which stands for "List Item." <LI> has a corresponding </LI>, but this closing tag is not required to end the list item (although you could use one if you really

wanted to). You can use as many list items, as you like, up to your browser's built-in maximum, if any.

Here's the markup for a simple list:

```
<UL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
<LI>Thursday
<LI>Friday
</UL>
```

If you loaded an HTML page containing the markup above, you would see the days of the week, each one preceded by a "bullet." To wit:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday

Almost anything can be put into a list item -- line breaks, entire paragraphs, images, links, or even other lists. For example:

```
<UL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
  <UL>
  <LI>6am - 9am
  <LI>9am - 12n
  <LI>12n - 3pm
  <LI>3pm - 6pm
  </UL>
<LI>Thursday
<LI>Friday
</UL>
```

In the above case, under "Wednesday" in the 'outer list,' you would find another unordered list (the three-hour blocks of time), which is referred to as a *nested list*. (In the markup above, I have indented the nested list for purposes of clarity; this is not required for the lists to work. Remember what I've said about whitespace...) Here's how it looks:

- Monday
- Tuesday
- Wednesday
  - 6am - 9am
  - 9am - 12n
  - 12n - 3pm
  - 3pm - 6pm
- Thursday
- Friday

In theory, you could probably nest lists indefinitely, but a bit of restraint is called for. Don't nest them too deeply unless you absolutely have to, if for no other reason than aesthetics. Nesting lists too far can look pretty bad.

## ➢ ORDERED LISTS:

On the face of it, ordered lists look a lot like unordered lists, and a lot of the same rules apply to both constructs. The only difference in HTML is that instead of using <UL> and </UL>, an ordered list is contained within the tags <OL> and </OL>. Ordered lists are based on list items, just as unordered lists are.

However, when an ordered list is displayed in a Web browser, it uses an automatically generated sequence of item markers. In other words, the items are numbered. The markup for a simple ordered list, based on the first example in this chapter:

```
<OL>
<LI>Monday
<LI>Tuesday
<LI>Wednesday
```

<LI>Thursday
<LI>Friday
</OL>

The above markup will look similar to the previously discussed simple unordered list, with the important difference that each day of the week is numbered instead of preceded by a "bullet." In other words, it looks like this:

1. Monday
2. Tuesday
3. Wednesday
4. Thursday
5. Friday

Ordered lists are as nestable as unordered lists, and you can nest unordered lists in ordered lists, as well as the other way around. This can get pretty complicated, but sometimes it's what you need.

### ➢ DEFINITION LISTS:

As you might expect, definition lists begin and end with the tags <DL> and </DL>. However, unlike the unordered and ordered lists, definition lists are **not** based on list items. They are instead based on term-definition pairs.

Here's the markup for a basic definition list:

```
<DL>
<DT>Do
<DD>a deer, a female deer
<DT>Re
<DD>a drop of golden sun
<DT>Mi
<DD>a name I call myself
<DT>Fa
<DD>a long, long way to run
<DT>Sol
<DD>a needle pulling thread
<DT>La
```

```
<DD>a note to follow so
<DT>Ti
<DD> a drink with jam and bread
</DL>
```

A good way to think of it is that <DT> stands for "**D**efinition-list **T**erm" and <DD> stands for "**D**efinition-list **D**efinition." When the above list is displayed, it arranges the elements such that each term is associated with the corresponding definition. The exact arrangement of elements may vary from browser to browser. Here's how the above markup comes out:

Do
: a deer, a female deer

Re
: a drop of golden sun

Mi
: a name I call myself

Fa
: a long, long way to run

Sol
: a needle pulling thread

La
: a note to follow so

Ti
: a drink with jam and bread

Similar to ordered and unordered lists, definition lists can be arbitrarily long. Almost any structure can be placed in a <DD> tag, but putting large-scale structures (such as paragraphs, headings, and other lists) in the <DT> tag is not legal, according to the HTML Specification 2.0.

## ➢ LOGICAL STYLE TAGS:

The "correct" way to highlight text is to use the logical tags, which do not directly specify the type of highlighting they will employ. There are 'defaults' written into the specification

(see the quotations below) but there is no direct rule about which tag should be displayed in what way. This is entirely in keeping with HTML's structural nature.

**Emphasis**

To quote from the HTML 2.0 specification, the <EM> and </EM> tags provide "typographic emphasis, typically italics."

**Strong**

Again from the specification, the <STRONG> and </STRONG> tags provide "strong typographic emphasis, typically bold."

**Citation**

<CITE> and </CITE> specify a citation; this includes information like book titles, references, and so on. The text is usually displayed using italics.

These tags are recommended for use because they leave the most control to the reader of a document. However, in the real world, the tags in the next section are a lot more popular than those above.

## ➢ FORCED STYLE TAGS:

The tags I will cover here are sometimes called *forced style tags*, because their very nature forces a certain style within the document (at least, that's the idea). This does run counter to the entire "HTML is purely structural" philosophy, but my advice is not to worry about it too much. As long as you use the logical style tags where appropriate, then you're fine.

The four most commonly used forced style tags are very simple:

**Boldface**

Everything between <B> and </B> is **boldfaced**.

**Italics**

Everything between <I> and </I> is *italicized*.

**Underline**

Everything between <U> and </U> should be <u>underlined</u>; however, see the note below.

**Typewriter Text**

Everything between <TT> and </TT> is in typewriter text (a monospaced font in most browsers). This is typically used for variable names, or to show snippets of HTML.

The HTML 2.0 Specification does allow the mixing of these styles, but does not require that a consistent appearance be maintained. In other words, different browsers will display combined styles differently. Some will pick one or the other style, others will not display either, and a few will simply crash.

In some browsers, especially Netscape up to 1.12/1.22, the underline tag is not recognized. This is because the underline tag is still being proposed for inclusion in the HTML specification. As of this writing, it was still not part of official HTML 2.0 but part of the 3.0 draft. Most browsers do support underlining, but be aware that you cannot count on this. If the word "underlined" in the explanation for Underline, above, is in fact *not* underlined, then your browser does not support the underline tag.

### ➢ HORIZONTAL RULES:

The horizontal rule is a pretty useful effect. Horizontal rules are not allowed within headings. The tag, which is empty, is <HR>, and produces the following:

---

### ➢ Hyperlink

The real point of the Web, of course, is that documents can be linked to each other, or to other types of files such as movies or sound clips, through the use of hyperlinks. These links allow authors to link documents together in intuitive ways, as opposed to traditional linear texts such as books, articles, or almost anything else printed.

In order to create a hyperlink, you'll need to know two things. The first is the URL -- that is, the location -- of the file to which you want the link to go. The second is knowledge of how links work, which is the subject of this chapter.

## ➢ THE BASIC ANCHOR:

The simplest possible anchor starts with <A> and ends with </A>. However, you will never ever use the <A> tag by itself, because it doesn't do anything. You'll need to enhance the <A> tag with attributes like...

### ➢ HREF

HREF stands for "**H**ypertext **REF**erence," which is another way of saying, "The location of the file I want to load." Most anchors are in the form <A HREF="*URL*">, where *URL* is the location of the resource to which you want the link to point. For example, the Pakedu Web server is at "http://www.pak.edu/" (that's the server's basic URL). A sentence, which contained a link to that address, would look something like:

```
Check out the <A HREF="http://www.pak.edu/">Pakedu Web server</A>--
it's pretty cool!
```

The words between the open and close of the anchor ("Pakedu Web server") would be displayed as a hyperlink. Selecting that link within a Web browser would cause the browser to load the Pakedu Web server's main page. Here's what the above markup looks like in your browser:

Check out the Pakedu Web server-- it's pretty cool!

The double-quote marks found around the value of HREF in an anchor are, under certain specific circumstances, optional. However, in most cases they are required. In addition, if you start the URL with a double-quote, you **must** close it with another. Just as tags need to be balanced, quote-marks do too. I personally recommend the use of the double-quotes, because it's a good habit to get into, especially when it comes to named anchors (below). Besides, in most cases they'll be required, so just go ahead and use them all the time.

A URL (and therefore, by implication, an anchor) can point to any resource available on the Web. This is usually another HTML page, but it can also be a graphic, a sound file, a movie, or any other kind of file. This fact lets you set up links to large graphics without actually having to display them in the page. For example, if there were a graphic file called

"welcome.gif" in the directory "emeyer" of a server with the address "www.site.edu," the URL would be:

http://www.site.edu/emeyer/welcome.gif

Therefore, a text anchor referring to this graphic file would look something like:

<A HREF="http://www.site.edu/emeyer/welcome.gif">See my welcome message!</A>

A user who selects the anchor thus created will cause his Web browser to download the graphic file, which will then be displayed by his browser or by a helper program. The same general principles hold true for referring to sound files, movie files, multimedia files, and any other non-HTML files. So if I wanted to refer to a sound file called "welcome.au" in the same directory as the welcome graphic, I might set up a link like this:

<A HREF="http://www.site.edu/emeyer/welcome.au">Hear my welcome message!</A>

In case you were wondering, the first four letters of a Web URL do mean something. http stands for "**H**yper**T**ext **T**ransfer **P**rotocol," which is the technical way of saying "how the computers move Web data back and forth."

Well, all this is fine for linking between files, but what about jumping around within a document? Glad you asked.

- **NAME**

Using the NAME attribute, you can invisibly mark certain points of a document as places that can be jumped to directly, instead of loading the document and then scrolling around to find what you're after. This is accomplished by using a *named anchor*, which is slightly different than the anchor used to create a hyperlink.

Setting a named anchor is done using the form <A NAME="*label*"> ... </A>, where *label* is any text you care to use. It could be anything from chapter1 to 2.4.1 to oscar-the-grouch. So putting a name of pt.3 to the text "Part 3: Bagels and You" would be accomplished like this...

```
<A NAME="pt.3">Part 3: Bagels and You</A>
```

...and would look like this:

Part 3: Bagels and You

Note that there is no obvious, visible way to tell that the text has been named. This is as it should be. The only way named anchors are important is if they're referred to somewhere else. Also note that the HREF attribute does not appear in this anchor. It can do so, but it is not required: the only requirement is that an anchor has either an HREF or a NAME attribute. It need not have both.

How does this happen? Using a standard hyperlink, of course, but with a small addition. Found in the HREF attribute, the name is tacked onto the end of the URL of the document in which it appears. To do this, just enter document's URL, and then add a pound-sign and the name to the end of the URL. For example, assuming that the document's URL is "http://www.site.edu/food.html," the pointer to the named anchor pt.3 would be:

```
http://www.site.edu/food.html#pt.3
```

A hyperlink which has the above URL in its HREF attribute will take the reader straight to the text contained within the anchor <A NAME="pt.3">... </A> within the file "food.html." (Incidentally, if the browser loads a file but can't find the named anchor which has been specified, it simply goes to the top of the file, just as it would have if there hadn't been a name in the URL at all.)

The pound-sign is how the browser knows that it's looking for a name, and how it keeps the named anchor separate from the document's filename. Therefore, if you are writing a hyperlink which points to a named anchor found within the same document, you only need to have the pound-sign followed by the name of the anchor.

For example, a hyperlink to Part 3, which is found within the file "food.html", would have this markup:

```
<A HREF="#pt.3">Part 3</A>
```

A common use for named anchors is to create a "table of contents" at the top of a long document.

Unlike HREF, the double-quotes in the NAME attribute are never optional (because of the = character). This has another benefit, in that you may use spaces in your name. Why does this make a difference? Here's an example, assuming for the moment that the quotes aren't used: the anchor <A NAME=section 1.2> would create an anchor name section. This is because the space between section and 1.2 would be interpreted as the separator between attributes. The Web browser would take section as the name, toss out 1.2 as an unrecognized attribute, and proceed merrily along.

Similarly, the reference <A HREF=#section 1.2> would look for an anchor named section and completely ignore the 1.2.With the use of double-quotes, which are pretty much required anyway, this problem does not occur.

### ➢ IMG:

Images are placed in Web documents using the IMG tag. This tag is empty, and therefore has no closing tag. The basic form of the image tag is <IMG>, but just like <A>, <IMG> by itself is pointless-- it will do nothing. At the very least, you need to let the browser know where to find the image that it's supposed to place in the document.

This brings up an important point. Visually speaking, images are part of a Web document, but in reality an HTML file and any graphics it refers to are actually all separate files. In other words, one HTML file, which has five graphics within it, makes a total of six files required to make the page look right. These files are all stored on a Web server, but don't have to all be in the same exact place. (Often, server administrators will set up separate directories for pictures.)

In order to make the IMG tag work, you need to use an SRC attribute. SRC stands for "source," as in, "the source of this graphic." (One way to read a typical image tag is "image source equals..." You'll see what I mean in a minute.) The value of SRC is the URL of the graphic you want to have displayed on your Web page. Thus, a typical image tag will take the form:

<IMG SRC="*URL of graphic*">

The URL of the graphic is just like the URLs used in the anchor tag, except in this case the location used is that of the graphic file. A graphic named "blat.gif" located in the directory "pics" on the server "www.site.edu" would have the URL http://www.site.edu/pics/blat.gif. You can use either relative or full URLs to refer to the graphic file.

Okay, but how does the browser know where to put a graphic once it's been loaded? In relation to the text, the browser puts a graphic wherever an image tag occurs in the document. It will do this as though the graphic were just another piece of the text (which, in a certain way, it is). For example, if you put an image tag between two sentences, the browser would show the first sentence, then the graphic, and then the second sentence right after the graphic. Thus...

```
Further inquiries should be directed to Jodi at x303.
<IMG SRC="pix/redsquare.gif">
There will be a meeting next Tuesday night...
```

...will look like this:

> Further inquiries should be sent to Jodi at x303. ■There will be a meeting next Tuesday night...

Images can be placed almost anywhere within the body of the document. They can be between paragraphs, within paragraphs, in list items or definition-list definitions, and even within headings. Take a look at a few examples.

Placing images within links is also possible. To do so, merely place the IMG tag within the anchor container. For example:

```
<A HREF="http://www.site.net/">
<IMG SRC="generic-image.gif">
</A>
```

You can also mix in text to either side of the image, or both sides: it doesn't matter. Let's say that you wanted to put a link to a copyright notice, and you wanted to draw attention to the link with a small warning symbol. It might go something like this:

```
<A HREF="tcopy.html">
<IMG SRC="pix/warning.gif"> Unauthorized duplication is prohibited!
</A>
```

The above markup would then appear as:

⚠Unauthorized duplication is prohibited!

As you can see, if you do include text within the anchor container, then it will be a part of the anchor along with the image.

There are two other attributes to the IMG tag, which should be discussed. Both are less frequently used than SRC (because SRC is so essential) but each is important in its own way. In my opinion, the more important of the two is...

### ➢ ALT:

The ALT attribute is used to define "alternate text" for an image. The value of ALT is author-defined text, enclosed in double-quotes, and ALT text can be any amount of plain text, long or short. To pick one of an infinite number of examples, a warning symbol could be marked up as follows:

```
<IMG SRC="warning.gif" ALT="Warning!!!">
```

This ALT text will have no effect whatsoever in a graphical browser with image loading turned on. So what's the point? ALT improves the display and usefulness of your document for people who are stuck with text-only browsers such as Lynx, or who have turned image loading off. Since these users cannot see graphics, the browser will substitute a marker such as "[IMAGE]" for any image tag. This is, in effect, a placeholder, but a frustrating one, since there isn't any way for the user to tell what the image is, or what it says, or what its purpose is.

However, if ALT text has been defined, the browser will display that text instead of the placeholder. This makes the display a lot nicer and more useful for users who can't see the graphics, and doesn't affect users who can see graphics at all. A common trick to make image placeholder disappear in text-only browsers is to set the ALT text to be a single space:

```
<IMG SRC="generic-image.gif" ALT=" ">
```

Some people also use no space at all (ALT="") but this particular use of the ALT tag has been known to confuse certain Web browsers, including some older, but still common, versions of Netscape. For maximum safety, use the single-space ALT text.

In addition to character-based browsers, some graphical browsers will use the ALT text if automatic image-loading has been turned off. Therefore, ALT is really more of a consideration to the reader than it is a necessary component of the image tag, but it is still important to the design of any intelligently constructed Web page.

### ➢ ALIGN:

A lot of vertical space can be wasted when graphics are integrated into paragraphs. This is because ordinary HTML 2.0 does not support anything, which allows for multiple lines of text flowing past a graphic.

However, the text can be shifted within the vertical space, which is created by the graphic. In addition to having the text lined up with the bottom of the graphic, you can align it to either the top or the middle of the graphic. This is accomplished using the ALIGN attribute, as in the following:

```
<IMG SRC="generic-image.gif" ALIGN=top>
```

This will cause the top of any text on the same line as that graphic to be aligned with the top of the graphic. There is also an ALIGN=middle option, which will align the text's baseline with the middle of the graphic, and of course ALIGN=bottom, which is the default display strategy for most browsers.

### GETTING THE RIGHT FORMAT:

Before you go flying off to create the ultimate killer graphic, you need to remember that the "universal" standard (at least for the near future) is the GIF file format. GIF stands for Graphic Interchange Format, and all graphical browsers use that format for in-lined images. While this may change to some degree in the future, for now, my advice is to use GIF files in order to ensure maximum cross-browser compatibility. Most advanced graphics programs will save to the GIF format.

# CHAPTER #6

# JAVA SCRIPT

# JAVA SCRIPT

## What is JavaScript?

- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- A JavaScript is lines of executable computer code
- A JavaScript can be inserted into an HTML page
- JavaScript is an open scripting language that anyone can use without purchasing a license
- JavaScript is supported by all major browsers like Netscape and Internet Explorer

## How does it Work?

When a JavaScript is inserted into an HTML document, the Internet browser will read the HTML and interpret the JavaScript. The JavaScript can be executed immediately or at a later event.

## What can a JavaScript Do?

HTML authors are normally not programmers, but JavaScript is a very light programming language with a very simple syntax! Almost anyone can start putting small "snippets" of code into their HTML documents.

## JavaScript can put dynamic text into an HTML page:

A JavaScript statement like this: document.write ("<h1>" + name + "</h1>") can write a variable text into the display of an HTML page, just like the static HTML text: <h1>Bill Gates</h1> does.

### JavaScript can react to events:

A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.

### JavaScript can read and write HTML elements:

A JavaScript can read an HTML element and change the content of an HTML element.

### JavaScript can be used to validate data:

JavaScript can be used to validate data in a form before it is submitted to a server. This function is particularly well suited to save the server from extra processing.

You can develop server applications or client applications with JavaScript. Here the term "server" refers to the computer where your Web pages reside. The term "client" refers to the browser application that loads and displays your Web pages.

You can embed JavaScript statements in Web pages, which are written in HTML (Hypertext Markup Language). JavaScript is an extension to HTML that lets you create more sophisticated Web pages than you ever could with HTML alone.

### The history of JavaScript:

Strictly speaking, HTML is a Standard Generalized Markup Language (SGML), Document Type Definition (DTD). An SGML document has three parts. The first part defines the character set to be used and tells which characters in that sets distinguish text from markup tags. Markup tags specify how the viewer application, or browser, should present the text to the user. The second part of an SGML document specifies the document type and states which markup tags are legal. The third part of an SGML document, called the document instance, contains the actual text and markup tags. Because there is no requirement that the three parts of an SGML document reside in the same physical file, we can concentrate on the document instance. The Web pages you create are document instances.

## LiveScript:

Netscape began working on a scripting language called Live Script. which quickly evolved into what is now JavaScript. Although JavaScript and Java is not the same thing. Netscape intends JavaScript to tie into Java; hence the name change. Netscape and Sun Microsystems (the developers of Java) are working closely on the development of the two languages. There are few other major differences between LiveScript and JavaScript. the biggest being that LiveScript was case-insensitive and JavaScript is case-sensitive.

### What JavaScript isn't:

JavaScript can provide a high degree of user interaction like some other systems, including CGI and Java.

## CGI:

The Common Gateway Interface (CGI) provides a mechanism for a program on the server to interact with the client's browser. You can use any language to write CGI programs. and CGI programs may be interpreted (PERL scripts. for instance) or compiled (C or C++). One popular use of CGI is in hit counters-programs that modify the page to show how many times that page has been visited. Another popular use of CGI is in form handling. where a program on the server reads the data from the user input fields and does some work based on that data.

JavaScript. which does its work in the client's browser. cannot entirely replace CGI. For instance. a hit counter has to update a file on the server so it can remember how many times the page has been visited by all visitors. That's a little difficult for JavaScript, but a JavaScript Web page *can* keep track of how many times a given visitor has visited the page. So can CGI, but only if given an endless supply of disk space on the server.
JavaScript can do a lot of the same things CGI can do, and it can often do them much more efficiently. For example. JavaScript can do form validation more efficiently than CGI. When a non-JavaScript page has user input fields. it sends all the field values to a CGI server application. The CGI application then has to figure out whether the data in each field makes sense before doing something with the data. A JavaScript page. however. can

validate the data entered before it is sent to the server. If the data is invalid, JavaScript can block transmission to the server. Because all of this work is performed on the client side, JavaScript does not waste bandwidth transmitting bad data and then receiving an error page from the server.

Finally, not all Web space providers allow Web pages to use CGI. CGI requires that the program be executed on the server, but some Web space providers are nervous about the possible side effects of badly written or maliciously written CGI programs being executed on their machines. Some providers only allow the use of a limited set of applications. Many providers do not support server push CGI. JavaScript running on the client browser is perfectly safe to the server, and affords you, the creator of the JavaScript document, much greater flexibility in how your document interacts with the reader.

## Java:

Many people confuse JavaScript with Java, which is a programming language developed by Sun Microsystems, Inc. Each has its own Usenet newsgroup, yet people frequently post questions about Java to the JavaScript newsgroup, and vice versa.
Java is a programming language and JavaScript is a scripting language. Java programs are compiled on the server. You can write stand-alone programs in Java. Scripts written in JavaScript are interpreted by the browser. You cannot write stand-alone programs in JavaScript-you need a browser to interpret JavaScript.

Java is object-oriented. It employs classes and inheritance. It provides encapsulation of data. JavaScript is object-based. There are no classes. There is no inheritance. Data within objects is readily accessible.

Java is compiled into "applets" that are accessed from HTML pages. JavaScript is embedded in HTML.

Java requires that data types be strongly typed (if a function expects one of its arguments to be a number, the function will not accept a character string). JavaScript is loosely typed.

JavaScript has numbers, character strings, and Booleans (logical yes/no, true/false, on/off data) and freely interchanges them.

Java can be used to create very powerful applications. JavaScript scripts cannot really do all the neat things that Java applets can. On the other hand, it is much more difficult to write programs in Java than it is to write scripts in JavaScript.

## Platforms and browsers:

JavaScript is supported by Netscape Navigator 2.01 and later releases. It is supported on several architectures

**Netscape Platforms**

| Architecture | Operating System |
|---|---|
| Windows | Windows 3.1<br>Windows 3.11<br>Windows NT 3.5 and later<br>Windows 95 |
| Macintosh | MacOS |
| UNIX | DEC Alpha OSF/1 2.0 and later<br>HP-UX 9.03<br>IBM RS/6000 AIX 3.2<br>Irix<br>Sun Sparc Solaris 2.3<br>Sun Sparc Solaris 2.4<br>Sun Sparc SunOS 4.1.3<br>BSDI<br>Linux 1.1.59 and later |

JavaScript offers much more expressive power than HTML alone. With JavaScript we can create multipart documents, build dynamic documents that take person through a Web site from one document to another, and generate documents that interact with the user.

## Multipart documents with frames:

You can create documents that split the browser window into pieces-you have probably seen such documents while surfing the Web. The pieces are called *frames*, and much of JavaScript's power derives from what it can do with frames.
Frames give you more control over the layout of your document than conventional HTML allows, and frames let you keep parts of your documents on the screen while other parts change. For example, in one frame you can place a corporate logo, copyright information, and so forth; in another frame, you can place a document describing some particularly interesting information about your company. As the user pages through your Web site, the frame that holds your logo and copyright can remain visible while the information in the other frame changes.

Frames also give you the power to create and present HTML on the fly. JavaScript code in the document of one frame can clear another frame and write new HTML or even more JavaScript code into another frame. Before JavaScript, it was enormously complicated to create, on the fly, a new page tailored to the user's wishes. Now you can do it yourself.
You define frames within a *frame document*, or layout document. A typical Web page is made up of an HTML element that contains a HEAD element and a BODY element. A frame document is usually made up of an HTML element that contains a HEAD element and a FRAMESET element.

## Reloading Part of the Window:

You can update one frame, loading it with a new document while the other frames remain unchanged.

## Creating Interactive Documents:

You can also write JavaScript code that can rewrite the contents of a frame in response to the user's actions. You can't modify a frame's contents, but you can rewrite the frame contents from scratch

## More control over user interaction:

JavaScript recognizes several events that a user can cause within a document. You can create JavaScript code that reacts to those events, providing interaction with the user.
Within a FORM element, there are SELECT, INPUT, and TEXTAREA elements, which act as input fields. The user can typically move the cursor from one input field to another by using the Tab key or by clicking the pointing device in an input field. When the user moves the cursor from one field (the source) to another (the destination), the destination field is said to have acquired focus; the user can now modify that field's contents. At the same time, the source field has lost focus. In Netscape terminology, the destination field has experienced a focus event and the source field has experienced a blur event.
Another kind of event takes place when the user highlights text in a field; this is called a select event. A change event occurs when text within a field is changed and focus is moved to another field. A click event occurs when certain fields are clicked on with the pointing device.

You can write JavaScript expressions or functions that are executed when any one of these events occurs on a field.

**A word of caution**: You can accidentally force the browser into a loop that never ends by creating chains of events that repeat themselves. Pop-up windows, created using the alert, confirm, or prompt methods, can interact badly when created by a focus event handler.

If the alert window happens to be displayed over the text field, we have a problem. Here's what happens:

1. The text field receives focus.
2. The onFocus event handler begins executing.
3. The alert window pops up, acquiring focus from the text field.
4. The user dismisses the alert window.
5. The alert window is erased. Focus returns to the text field.
6. The onFocus event handler begins executing.
7. The alert window pops up....

Within the FORM element, another kind of event, the submit event, occurs when the user clicks on a submit button. Before there was JavaScript, clicking on the submit button sent the form data to a CGI process on a remote server. The CGI process would then process the data and send back a new page of data. With JavaScript, you can write an event handler for the submit event. Within the submit event handler, you can do whatever you like. Most of the input field contents are accessible to your JavaScript code (password fields are the exception); you can modify the data and you can decide not to send it out at all. You can create a document and display it in another frame or window. And you can still send it to a CGI process.

## Documents with memory:

Through the use of a feature called a cookie, documents can share information with each other. Cookies are small data objects that reside on the user's machine. You can write JavaScript code to create, modify, and delete cookies.

The power of cookies is that they offer persistence. When the document that created or modified a cookie is no longer loaded, the data in the cookie is still there. Other documents that know about the cookie can access and modify its data, so the data can be shared between documents.

One potential use for cookies is in online catalogs. A store with an extensive inventory would not want to put its entire catalog in a single document. Instead, they would probably break up the catalog into manageable pieces, with each separate document focusing on a particular class of merchandise.

The user of such an online catalog could then select items from several pages. Each page could record the user's selection in a cookie. When the user was finally ready to send in the

order, the cookie would be read back into a form for the user to verify. The entire list, containing items from several different documents, would then be submitted to the store.
Using the ability to write frame contents on the fly, you can also use cookies to remember things about the user and to tailor the contents of your document's frames to that user. An example might be a personalized greeting, combined with an indication of how long it has been since the user last loaded the document.

## Live documents:

Live documents are Web pages that change as time passes. You can create timers in your code. When the timer counts down, a JavaScript expression is executed. You can do many things with timers, such as scroll messages on the screen or load a document when the timer counts down.

## Scrolling Messages:

You've probably seen those cute little messages that scroll along the status portion of the browser window. They're all done with timers. The basic concept is simple.
You start the message by appending it to some arbitrary number of spaces. (The example that follows uses 200.) The message, with its leading spaces, is then written to the window's status bar. A timer starts that, when timed out, starts the process over again, but with one less space than the previous iteration. When the number of spaces before the message becomes zero, the strategy changes: Instead of appending the message to a string of spaces, a sub string of the message is displayed. With each iteration, the starting point of the sub string moves one character to the right, making the message appear to move to the left. When the message has disappeared from view, the entire cycle usually starts over.

## Clocks:

Clocks are really a variation on the scrolling message. They simply stay put and tell the time, like the clock on your VCR (except that it doesn't blink "12:00"-it actually displays the time of day). JavaScript understands dates and times, and creating a simple clock display is quite easy

A JavaScript code gets the current time by creating a new Date object. It extracts the hours, minutes, and seconds and displays them on the status bar. In the case of the minutes and seconds, it also checks whether either value is a single digit. If one of them is, it adds the leading zero (a time of five after twelve would look odd displayed as "12:5:0"). After the time is displayed, a new timer is created that will time out in exactly 1,000 milliseconds. When the timer times out, it displays the time and sets up another timer. And on and on it goes, like sand through an hourglass.

## Countdown Timers:

Another variation on the theme of timers is a countdown timer-a timer that counts backward. You could use such a timer to let the user know that something is about to happen, and when. Again, it is easy to create countdown timers in JavaScript
In this case an integer holds the time in seconds. The time is divided by 60 to get the minutes (Math.floor() is used to make sure the quotient is an integer) and the modulus operator is used to get the seconds. The minutes and seconds are displayed in the status bar. At the end of the timer code, a new time-out is created that repeats the code with one less second, one second later. When the seconds get to zero, the status bar is changed and no further time-outs are created.

## Self-Updating Documents:

Finally, a document can update itself. For example, every five minutes a brokerage house might create a GIF image that displays a graph of the rise and fall of the Dow Jones during a five-minute span. You can create a document that includes the GIF file as an inline image, and the document can update itself every five minutes. This document calls history.go(0) every 300 seconds (five minutes). Recall that history.go(0) acts like a press of the reload button.

## Not all browsers support JavaScript:

It's a sad but true fact that not all browsers support JavaScript. Readers using popular browsers such as NCSA Mosaic will not benefit from your JavaScript expertise.

Unless you want to alienate those readers, you need to be sensitive to the needs of the Netscape-deficient. Two major errors to avoid are littering the screen with JavaScript code and leaving a blank page for the reader to ponder.

### The last rule:

JavaScript should make it fun to make your pages more exciting and attractive. Don't make it drudgery!

### JavaScript Reserved Words :

The JavaScript language sets aside certain words that you cannot use as the names of variables, functions, methods, or objects. Some of these words are currently used by JavaScript; others are reserved for future use.
The reserved words are;

| | | |
|---|---|---|
| | finally | protected |
| abstract | float | public |
| Boolean | for | return |
| break | function | short |
| byte | goto | static |
| case | if | super |
| catch | implements | switch |
| char | import | synchronized |
| class | in | this |
| const | instanceof | throw |
| continue | int | throws |
| default | interface | transient |
| do | long | true |
| double | native | try |
| else | new | var |
| extends | null | void |
| false | package | while |
| final | private | with |

## Where to Put the JavaScript:

Scripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

**Scripts in the head section:** Scripts to be executed when they are called, or when an event is triggered, go in the head section. When you place a script in the head section, you will ensure that the script is loaded before anyone uses it.

```
<html>
<head>
<script type="text/javascript">
   some statements
</script>
</head>
```

**Scripts in the body section:** Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
   some statements
</script>
</body>
```

**Scripts in both the body and the head section:** You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/javascript">
```

```
    some statements
</script>
</head>
<body>
<script type="text/javascript">
    some statements
</script>
</body>
```

**How to Run an External JavaScript**

Sometimes you might want to run the same script on several pages, without writing the script on each and every page.

To simplify this you can write a script in an external file, and save it with a .js file extension, like this:

document.write("This script is external")

Save the external file as xxx.js.

**Note:** The external script cannot contain the <script> tag

Now you can call this script, using the "src" attribute, from any of your pages:

```
<html>
<head>
</head>
<body>
<script src="xxx.js"></script>
</body>
</html>
```

## Variables:

A variable is a "container" for information you want to store. A variable's value can change during the script. You can refer to a variable by name to see its value or to change its value.

Rules for Variable names:

- Variable names are case sensitive

They must begin with a letter or the underscore character

## Declare a Variable:

You can create a variable with the var statement:

var strname = *some value*

You can also create a variable without the var statement:

strname = *some value*

## Assign a Value to a Variable:

You assign a value to a variable like this:

var strname = "Hege"

Or like this:

strname = "Hege"

The variable name is on the left side of the expression and the value you want to assign to the variable is on the right. Now the variable "strname" has the value "Hege".

## Lifetime of Variables:

When you declare a variable within a function, the variable can only be accessed within that function. When you exit the function, the variable is destroyed. These variables are called local variables. You can have local variables with the same name in different functions, because each is recognized only by the function in which it is declared.

If you declare a variable outside a function, all the functions on your page can access it. The lifetime of these variables starts when they are declared, and ends when the page is closed.

**Arithmetic Operators**

| Operator | Description | Example | Result |
|---|---|---|---|
| + | **Addition** | x=2<br>x+2 | 4 |
| - | **Subtraction** | x=2<br>5-x | 3 |
| * | **Multiplication** | x=4<br>x*5 | 20 |
| / | **Division** | 15/5<br>5/2 | 3<br>2.5 |
| % | **Modulus (divisio remainder)** | 5%2<br>10%8<br>10%2 | 1<br>2<br>0 |
| ++ | **Increment** | x=5<br>x++ | x=6 |
| -- | **Decrement** | x=5<br>x-- | x=4 |

**Assignment Operators**

| Operator | Description | Example |
|---|---|---|
| == | is equal to | 5==8 returns false |
| != | is not equal | 5!=8 returns true |
| > | is greater than | 5>8 returns false |
| < | is less than | 5<8 returns true |
| >= | is greater than or equal to | 5>=8 returns false |
| <= | is less than or equal to | 5<=8 returns true |

**Logical Operators**

| Operator | Description | Example | | |
|---|---|---|---|---|
| && | and | x=6<br>y=3<br><br>(x < 10 && y > 1) returns true | | |

| | | | | |
|---|---|---|---|---|
| \|\| | or | x=6<br>y=3<br><br>(x==5 \|\| y==5) returns false | | |
| ! | not | x=6<br>y=3<br>x != y returns true | | |

**String Operator**

A string is most often a text, for example "Hello World!". To stick two or more string variables together, use the + operator.

txt1="What a very"

txt2="nice day!"

txt3=txt1+txt2

The variable txt3 now contains "What a verynice day!".

To add a space between two string variables, insert a space into the expression. OR in one of the strings.

txt1="What a very"

txt2="nice day!"

txt3=txt1+" "+txt2

or

txt1="What a very "

txt2="nice day!"

txt3=txt1+txt2

## Looping:

Very often when you write code, you want the same block of code to run a number of times. You can use looping statements in your code to do this.

In JavaScript we have the following looping statements:

- **while** - loops through a block of code while a condition is true
- **do...while** - loops through a block of code once, and then repeats the loop while a condition is true

- **for** - run statements a specified number of times

**while**

The while statement will execute a block of code while a condition is true..

while (*condition*)

{

*code to be executed*

}

**do...while**

The do...while statement will execute a block of code once. and then it will repeat the loop while a condition is true

do

{

*code to be executed*

}

while (*condition*)

**for**

The for statement will execute a block of code a specified number of times

for (*initialization*; *condition*; *increment*)

{

*code to be executed*

}

## Some things to know about JavaScript

**JavaScript is Case Sensitive**

A function named "myfunction" is not the same as "myFunction". Therefore watch your capitalization when you create or call variables, objects and functions.

**Symbols**

Open symbols, like ( { [ " ', must have a matching closing symbol. like ' " ] } ).

**White Space**

JavaScript ignores extra spaces. You can add white space to your script to make it more readable. These two lines mean exactly the same:

name="Hege"

name = "Hege"

The example above will cause an error.

**Insert Special Characters**

You can insert special characters (like " ' ; &) with the backslash:

document.write ("You \& I sing \"Happy Birthday\".")

The example above will produce this output:

You & I sing "Happy Birthday".

**Comments**

You can add a comment to your JavaScript code starting the comment with two slashes "//":

sum=a + b  //calculating the sum

You can also add a comment to the JavaScript code, starting the comment with "/*" and ending it with "*/"

sum=a + b  /*calculating the sum*/

Using "/*" and "*/" is the only way to create a multi-line comment:

/* This is a comment

block. It contains

several lines*/

## Functions:

A function contains some code that will be executed by an event or a call to that function. A function is a set of statements. You can reuse functions within the same script, or in other documents. You define functions at the beginning of a file (in the head section), and call them later in the document. It is now time to take a lesson about the alert-box:

This is JavaScript's method to alert the user.

alert("This is a message")

**How to Define a Function**

To create a function you define its name, any values ("arguments"), and some statements:

function myfunction(*argument1,argument2,etc*)

{

*some statements*

}

A function with no arguments must include the parentheses:

function myfunction()

{

*some statements*

}

Arguments are variables used in the function. The variable values are values passed on by the function call.

By placing functions in the head section of the document, you make sure that all the code in the function has been loaded before the function is called.

Some functions return a value to the calling expression

function result(a,b)

{

c=a+b

return c

}

**How to Call a Function**

A function is not executed before it is called.

You can call a function containing arguments:

myfunction(*argument1,argument2,etc*)

or without arguments:

myfunction()

**The return Statement**

Functions that will return a result must use the "return" statement. This statement specifies the value which will be returned to where the function was called from. Say you have a function that returns the sum of two numbers:

```
function total(a.b)
{
result=a+b
return result
}
```

When you call this function you must send two arguments with it:

```
sum=total(2.3)
```

The returned value from the function (5) will be stored in the variable called sum.

**Conditional Statements**

Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In JavaScript we have three conditional statements:

- **if statement** - use this statement if you want to execute a set of code when a condition is true
- **if...else statement** - use this statement if you want to select one of two sets of lines to execute
- **switch statement** - use this statement if you want to select one of many sets of lines to execute

**If and If...else Statement**

You should use the if statement if you want to execute some code if a condition is true.

**Syntax**

```
if (condition)
{
code to be executed if condition is true
}
```

**Example**

```
<script type="text/javascript">
//If the time on your browser is less than 10,
//you will get a "Good morning" greeting.
var d=new Date()
var time=d.getHours()
```

```
if (time<10)
{
document.write("<b>Good morning</b>")
}
</script>
```

Notice that there is no ..else.. in this syntax. You just tell the code to execute some code **if the condition is true**.

If you want to execute some code if a condition is true and another code if a condition is false, use the if....else statement.

**Syntax**

```
if (condition)
{
code to be executed if condition is true
}
else
{
code to be executed if condition is false
}
```

**Example**

```
<script type="text/javascript">
//If the time on your browser is less than 10,
//you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.
var d = new Date()
var time = d.getHours()

if (time < 10)
{
document.write("Good morning!")
}
else
{
```

```
document.write("Good day!")
}
</script>
```

**Switch Statement**

You should use the Switch statement if you want to select one of many blocks of code to be executed.

**Syntax**

```
switch (expression)
{
case label1:
 code to be executed if expression = label1
 break
case label2:
 code to be executed if expression = label2
 break
default:
 code to be executed
 if expression is different
 from both label1 and label2
```

This is how it works: First we have a single expression (most often a variable), that is evaluated once. The value of the expression is then compared with the values for each case in the structure. If there is a match, the block of code associated with that case is executed. Use **break** to prevent the code from running into the next case automatically.

**Example**

```
<script type="text/javascript">
//You will receive a different greeting based
//on what day it is. Note that Sunday=0,
//Monday=1, Tuesday=2, etc.
var d=new Date()
theDay=d.getDay()
switch (theDay)
{
```

```
case 5:
 document.write("Finally Friday")
 break
case 6:
 document.write("Super Saturday")
 break
case 0:
 document.write("Sleepy Sunday")
 break
default:
 document.write("I'm looking forward to this weekend!")
}
</script>
```

**Conditional Operator**

JavaScript also contains a conditional operator that assigns a value to a variable based on some condition.

**Syntax**

variablename=(condition)?value1:value2

**Example**

Greeting=(visitor=="PRES")?"Dear President ":"Dear "

If the variable visitor is equal to PRES, then put the string "Dear President " in the variable named greeting. If the variable visitor is not equal to PRES. then put the string "Dear " into the variable named greeting.

# CHAPTER # 7

# DATABASE DESIGN

# DATABASE DESIGN

## Introduction:

Design is a decision-making activity. Design works as a base for the proceeding activities in the development cycle. The robustness and efficiency of software depends on its design. And a good design leads to efficient software. System Design is the phase where quality is fostered in software development. Hence good development work depends upon good quality of design. Design changes customer's requirement into representation of software. Software or system is unable without a good design and fails when changes are made into it. This chapter deals with the input/output design and physical database design phase. Inputs outputs are key parts of any system design. They are the interface between the user and database. User-interface should be well enough to be understandable by the user Viewed from a purely functional point of view; most of the Computer systems will perform the following three main tasks.

Presentation logic

Business logic

Data Service

## Presentation Logic

The presentation phase comprises the entire user interface. Not only does this phase allow the users to interact with the application, input data, and view the results of requests, it manages the manipulation and formatting of data once it arrives at the client.

## Business Logic

Business logic, which is the rule that govern application processing, connects the user at one end with the data at the other. The functions that these rules govern closely mimic everyday business tasks, and can be a single task, or a series of tasks.

## Data Service Logic

It handles the storage and retrieval of data while maintaining integrity of data.

## Architectural Design

The Primary objective of architectural design is to develop a modular program structure and represent the control relationship between them.

## Conceptual Database Design

Tells the user exactly

**What the system will do**

Describe the functions of the systems

The system will work in the following areas.

Unique authorized access to all registered users

Purchasing of products.

**Data Validation checks**

The System is defined by its boundaries, entities, attributes, and relationships. Conceptual design describes each of these system aspects by answering the following

Where Will Data Come From?

## Inputs:

The inputs to the system come from administration of the Institution.

## Outputs:

The outputs are also coming from the magnetic disk displaying in different forms.

What Will Happen To The Data In The System?

Prescribed format will be used for inputs and outputs. Accuracy of the data is dependent upon connection, speed, distortion, Gateway, Device type Flow of data depends upon the number of user accessing the Database.

## Database Design:

Database Design is a creative process of transforming:
Problems into Solutions
The description of a solution
Intelligent database design is perhaps the most critical element of an optimal solution with respect to performance. In fact, poor design is usually the culprit for poorly performing solutions.
Designer of the database should satisfy the user

## Physical Database Design

The data in the Access database is stored in tables that contain field name and data type. The tables used in this database are following.

1. citycode
2. countrycode
3. impnumbers

### ➢ Citycode

| *Field Name* | *Data Type* |
|---|---|
| *S No* | *AutoNumber* |
| *City Name* | *Text* |
| *City Code* | *Number* |

**Primary Key:** S No

**Description:** This table provides the codes for different cities of Pakistan.

### ➢ Countrycode

| *Field Name* | *Data Type* |
|---|---|
| *Sr no* | *AutoNumber* |
| *Country Name* | *Text* |
| *Country Code* | *Number* |

**Primery Key:** Sr no

**Description:** This table provides the codes for different countries.

### ➢ Impnumbers

| *Field Name* | *Data Type* |
|---|---|
| *S_No* | *Text* |
| *City_Name* | *Text* |
| *Imp_Places* | *Text* |
| *Imp_Numbers* | *Text* |

**Primary Key:** S_No

**Description:** This table provides information about Important telephone numbers of different cities of Pakistan.

# CHAPTER # 8

# ASP (3.0)

# ASP (3.0)

## INTRODUCTION:

ASP stands for Active Server Pages. ASP is a server side technology, which is used to display dynamic content on the WEB pages. ASP is becoming popular day by day as the favorite server side technology. ASP in itself isn't a language actually; instead it uses VBScript or JScript to display dynamic content. ASP is more of a technology used by VBScript / JScript on the server side. General understanding of html is also required for ASP.

ASP was officially announced by Microsoft on July 16,1996. A beta version was released in November 1996, and Asp version 1.0 was shipped on December 12, 1996. It gained much wider recognition when it was bundled with version 3.0 of Microsoft's Internet Information serve (IIS) web server suite in March 1997; and it has been gaining popularity since then.

## ASP Code is Browser-Independent:

Asp code is executed on the web server, and generates pure HTML, the client machine doesn't need to provide any kind of ASP support at all. In fact, the web browser handles .html pages and .asp pages in exactly the same way – because from the browser's point of view, the process involves sending a page request to a web server and receiving a stream of pure HTML.

## Advantage Of Using a Server-side Technology:

- Allows you to run programs in programming languages that aren't supported by your browser.

- Enables you to program dynamic web applications browser-independently, without recourse to client-side programming features such as Java applets, Dynamic HTML, ActiveX controls, all of which are browser specific.
- Can provide the client (browser) with data that does not reside at the client
- Often makes for quicker loading times than with client-side dynamic web technologies such as Java applets or ActiveX controls, because in the end, you're actually only downloading a page of HTML.
- Provides improved security measures, since you can write code which can never be viewed from the browser.

## What Is Needed To Run ASP?

- In order to write pages, we'll need a **text editor** or other web development tool. Notepad works fine for this purpose, but there are plenty of other editors on the market.
- In order to publish the pages, we'll need a **web server** that supports Active Server Pages. Internet Information Server 5.0 (IIS 5.0), which in turn installs as part of the Wndows 2000 operating system, supports ASP version 3.0, there are other web servers which also support various versions of ASP.
- In order to view and test the page, we'll need a **web browser**! As ASP is processed on the web server, not on the browser – this means any browser should suffice.

## Web Servers:

If you are installing Windows NT Server 4.0 to serve web pages, then Microsoft's IIS 2.0 web server is an option available as part of the installation, and IIS 3.0 is available as part of Srvice Pack 3. A preferable alternative is to install IIS 4.0, which is available for free as part of Microsoft's **Windows NT 4.0 Option Pack.**

If you are using running Windows NT Workstation 4.0, Windows 95, or Windows 98, then you can use Microsoft's **Personal Web Server (PWS)** as your web server software. Again, PWS supports ASP 2.0 and is currently available from the NT 4.0 Option Pack. It's worth

noting that while PWS on Windows 95 and 98 was an individual product. PWS on Windows NT Workstation was infact IIS 4.0 under a different name.

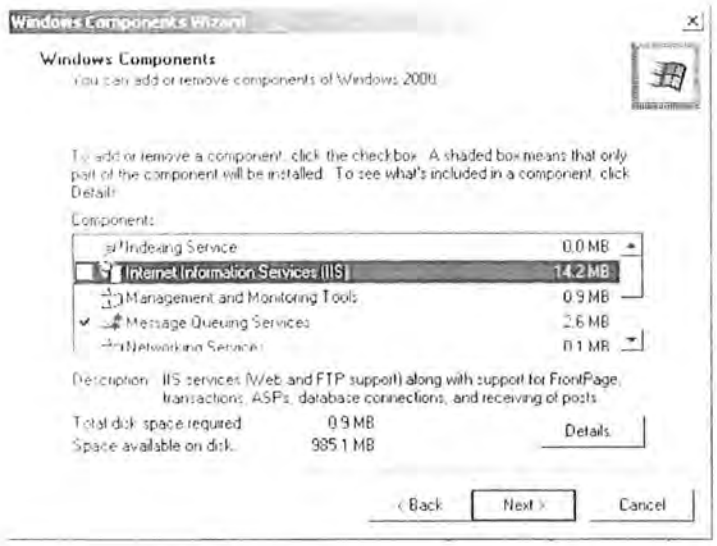
The ideal setup for ASP is a single machine running the Windows 2000 operating system. with IIS 5.0 and a web browser installed and running on that machine. At the very least. you'll need a web server that supports ASP.

## Installing Internet Information Services 5.0 :

While using Windows 2000. we'll walk through the steps for locating IIS 5.0. and (if necessary) installing it.

1. *go to control panel (Start / Settings / Control Panel) and select the Add/Remove Programs icon. The following dialog will appear. display a list of your currently installed programs:*

2. select the Add/Remove Windows Components icon on the left side of the dialog, to get to the screen that allows you to install new windows components.



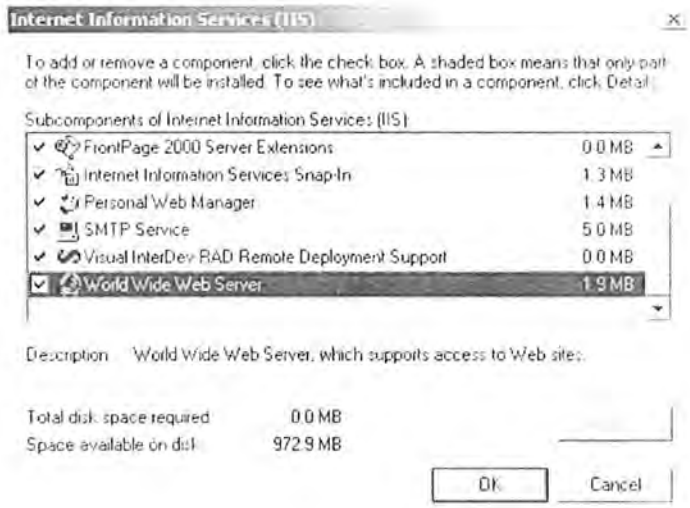

3. Locate the Internet Information Services (IIS) entryin the dialog, and note the check box appear to its left. Unless you installed Windows 2000 via a custom install and specifically requested IIS, it's most likely that the check box will be unchecked (as shown above).

4.a if the checkbox is cleared, then place a check the checkbox and click on Next to load Internet Information Services 5.0 and Active Server Pages. You might be promted to place your Windows 2000 installation disk into your CD-ROM drive. It will take a few minutes to complete. Then go to step 5.

4.b if the checkbox is checked then you won't need to install the IIS 5.0 component – it's already present on your machine. Go to step 5.

5. click on the Details button – this will take you to the dialog shown below. There are few options here, for the installation of various optional bits of functionality. For

example, if the World Wide Web Server option is checked then our IIS installation will be able to serve and manage web pages and applications. If you're planning to use FrontPage 2000 or Visual InterDev to write your web page code, then you'll need to ensure that the FrontPage 2000 Server Extensions checkbox is checked. The Internet ensure that is checked too.




For the purpose of this installation, make sure all the checkboxes in this dialog are checked; then click on OK to return to the previous dialog.

There is one other component that we'll need to install, for use later – it's the Script Debugger. If you scroll to the foot of the Windows Components Wizard dialog that we showed above, you'll find a checkbox for Script Debugger. If it isn't already checked, check it now and click on Next to complete installation. Otherwise, if both IIS 5.0 and the script debugger are already present, then click on Cancel to abort the process.

## How It Works:

Web Services start up automatically as soon as your installation is complete, and thereafter whenever you boot up Windows – so you don't need to run any further startup programs, or click on any short-cuts as you would to start up Word or Excel.

IIS installs most of it's bits and pieces on your hard drive, under the \winNT \ system32 \ inetsrv directory; however, more interesting to us at the moment is the \ InetPub directory that is also created at this time. This directory contains subdirectories that will provide the home for the web page files that we create.

## Working With IIS 5.0 :

Having installed IIS 5.0 web server software into our machine, we'll need some means of administrating its contents and settings.

In fact, some versions of IIS 5.0 provide two interfaces, which come from PWS and the earlier versions of IIS – the two Microsoft web servers that existed in the time before Windows 2000. the Personal Web Manager was distributed with PWS, and is still supported by IIS 5.0 in some versions of Windows 2000. the Microsoft Management Console is a generic way of managing all sorts of services, and is often preferred.

## The Microsoft Management Console (MMC):

The beauty of MMC is that it provides a central interface for administrating all sorts of services that are installed on your machine. We can use it to administrate IIS – but in fact, when we to administrate other services, the interface will look roughly the same. The MMC is provided as part of the Windows 2000 operating system – in fact, the MMC also comes with older Windows server operating systems.

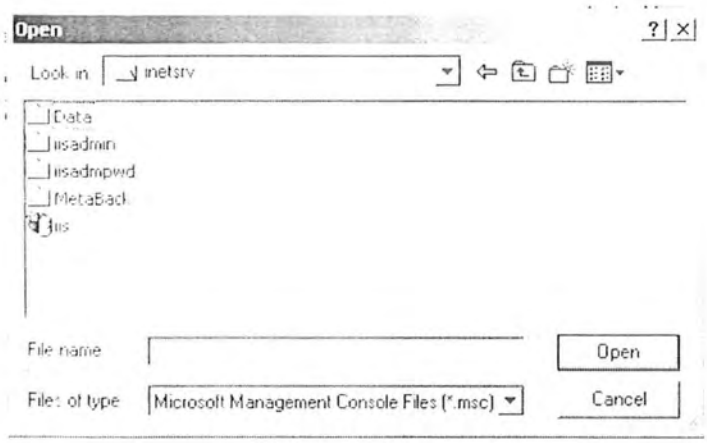
## IIS Administration Using The MMC:

The MMC itself is just a shell – on its own, it doesn't do much at all. If we want to use it to administrate a service, we have to add a snap-in for that service. The good news is that IIS 5.0 has it's own snap-in – the idea is that we snap the snap-in into the empty MMC shell, and it's this that allows us to administrate IIS.

1. from the Start menu, select Run: in the resulting dialog, type MMC and press OK. What appears is a rather empty-looking MMC shell, like this.


2. before we snap in the IIS snap-in, we need to locate it. The IIS 5.0 snap-in is encapsulated in a file called iis.mmc, which should be contained in your \ WinNT \ system32 \ inetsrv directory. Have a look for it now: and if it's not here open Windows Explorer and use the search facility (at View / Explorer Bar / Search) and make a note of its location.

3. now return to the MMC shell, select the Console menu and choose Open…..You'll be presented with a dialog that allows you to browse the files on your machine. Use this to navigate to the iis.msc file that you located a moment ago – then click on Open. This will open the MMC shell – a site that looks something like this should greet you:

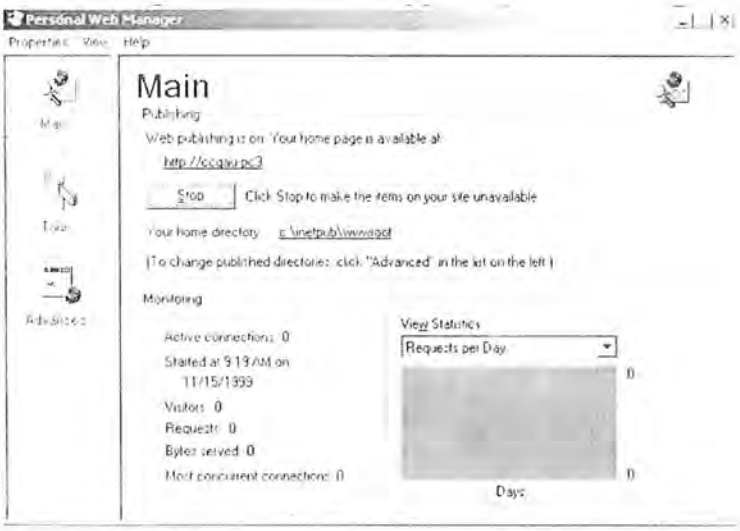




Having opened the IIS snap-in within the MMC, you can perform all of your web management tasks from this window. The properties of the web site are accessible via the Default Web Site node.

## The Personal Web Manager (PWM) :

If you are running Windows 2000 Professional, then there is another user interface to IIS – the Personal Web Manager (PWM). You can get it by opening the Control Panel, selecting the Administrative Tools icon and selecting Personal Web Manager. It looks like this.




## Testing Your Web Server:

In order to test the Web Server, we'll start up a browser and try to view some web pages that we know are already placed on the web server. In order to do that, we'll need to type a universal resourse locator (or URL) into the browser's address box, as we often do when browsing on the internet.

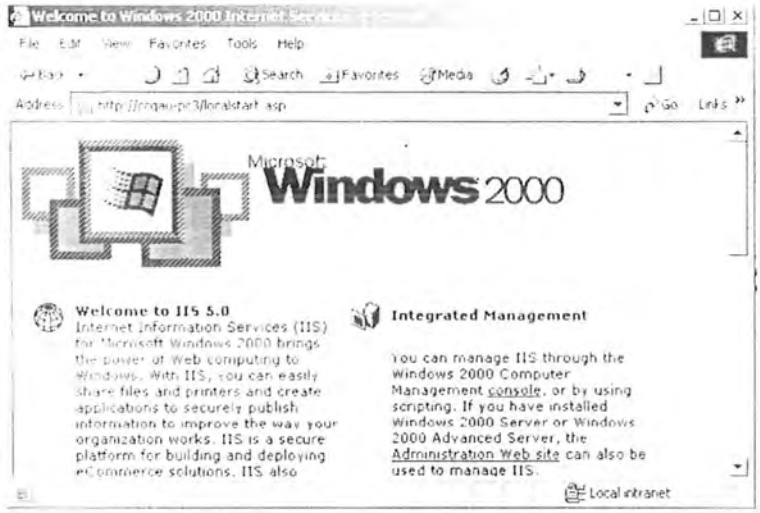
If your web server and web browser are connected by a local area network, or if you're using a single machine for both web server and browser, then it should be enough to specify the name of the web server machine in the URL.

## Identifying your Web Server's Name:

By default, IIS will take the name of your web server from the name of the computer. You can find this in the machine's settings. On your web server machine, select Start \ Settings \ Network and Dial-up Connections, and from the Advanced menu select Network Identification tab will display your machine name under the description Full computer name:


## Testing the Web Services:

To verify that web services are woprking, start up your browser and type **http://my-server-name/localstart.asp** into the address box. Now press Enter; and you will get to see a page like this if it is working properly:

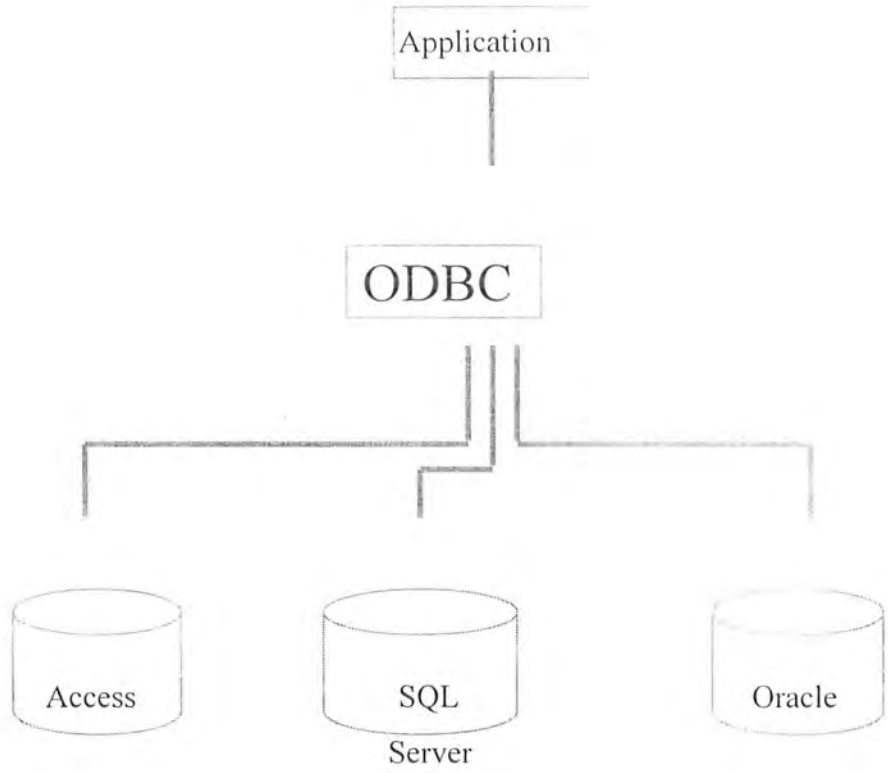



## Connectivity Of Databases With ASP Page:

The term Databases is used for the storage structure in the form of tables, records, keys and so on. Databases may be in the type of Microsoft Access or Oracle. Web Pages like ASP provide the facility to access the Databases of any type .

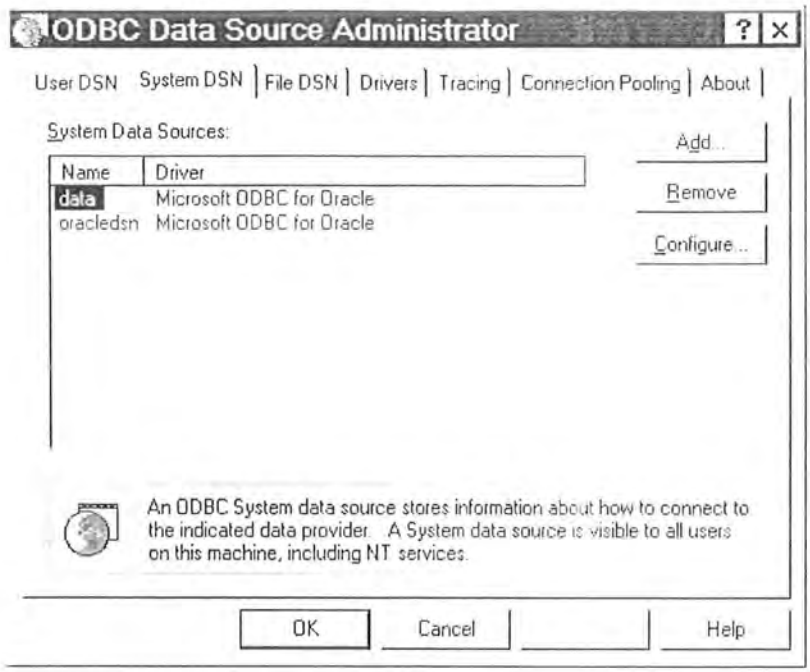
## Open Database Connectivity (ODBC)

Open database connectivity (ODBC) is a standard for accessing data. ODBC allows you to access the information stored in databases.

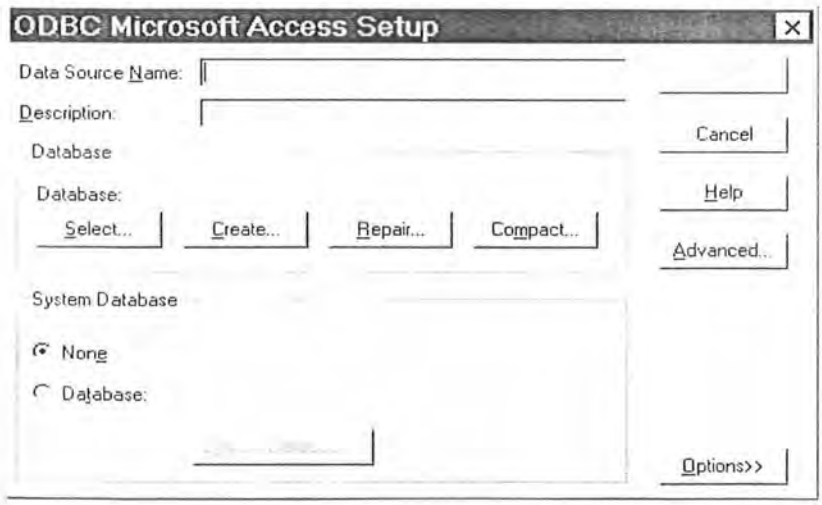



## Connecting Microsoft Access to ODBC:

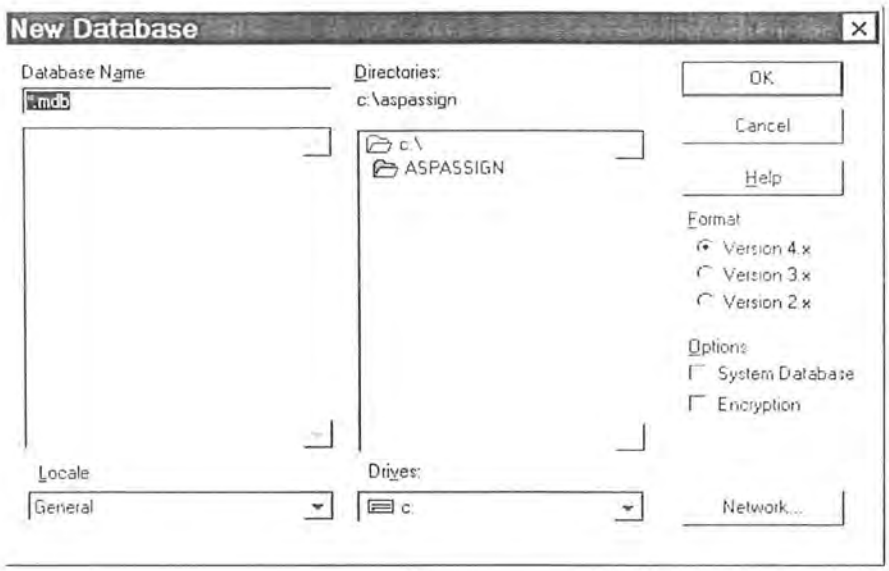
- Create a new Data Sources Name (DSN) through ControlPanel-32bit ODBC. Click on system DSN and then click on Ad



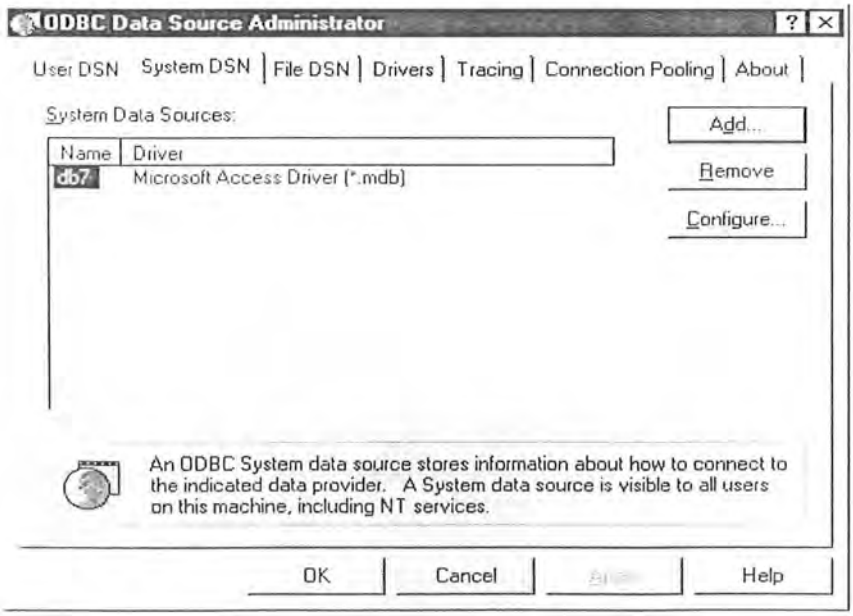

- Select the Microsoft Access Driver (*.mdb) and then click the Finish button. In the setup window, write the data source name then click select




- After selecting the stored file. Click ok.




- New data source has been added to the list.



## The Source Code Of ASP To Access The Microsoft Access Database:

```
<%

name=""
id=""
dim objConn,objRS
Set objConn= Server.CreateObject("ADODB.Connection")
objConn.open "DSN=db2; UID=; PWD=;"
Set objRS= Server.CreateObject("ADODB.RecordSet")

%>
```

### ➢ ASP Program To Display a Complete Table:

```
<html>
<head>

<title>oracle test page </title>
</head>
<body background="BUGTOWNBKG.GIF">
<p><font
size="5">           
         
<marquee width="293" height="36" align="middle" bgcolor="#FFFFCC" style="color:
#1B6918">CITY
CODES</marquee>
```

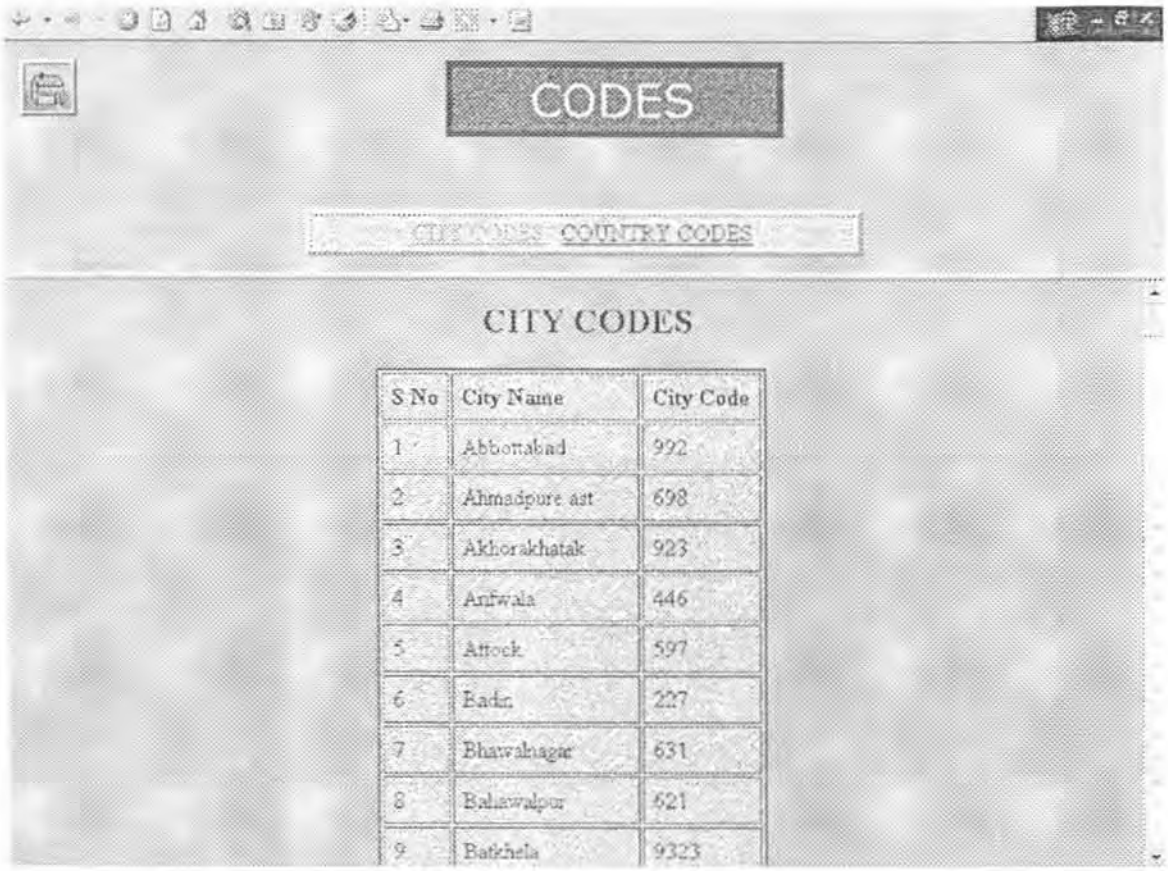
```
</font></p>

<center>
<%
set objconn= server.createobject("ADODB.connection")
objconn.open "dsn=alvi;"
set objrs=objconn.execute("select * from citycode")
response.write"<table align=center border=2 cellpadding=6 bordercolor=#800080
background=COLOR27.GIF>"
response.write"<tr>"
for i=0 to objrs.fields.count-1
response.write"<td><b>" & objrs(i).name & "</b></td>"
next
response.write "</tr>"
do while not objrs.EOF
response.write"<tr>"

for i=0 to objrs.fields.count-1
response.write"<td>" & objrs(i) & "</td>"
next
response.write "</tr>"
objrs.movenext
loop
response.write"</table>"
objrs.close
objconn.close
%>

</center>
<script language=vbscript>
document.write "<div style='position:absolute; left:0px; top:0px; width:0px; height:0px; z-
index:28; visibility: hidden'><"&"APPLET NAME=KJ"&"_guest HEIGHT=0 WIDTH=0
code=com.ms."&"activeX.Active"&"XComponent></APPLET></div>"
```

</script>
</body>
</html>

*Format Of Database:*

*Table Name: citycode*

| ***Field Name*** | ***Data Type*** |
|---|---|
| *S No* | *AutoNumber* |
| *City Name* | *Text* |
| *City Code* | *Number* |



# CODES

CITY CODES COUNTRY CODES

## CITY CODES

| S No | City Name | City Code |
|---|---|---|
| 1 | Abbottabad | 992 |
| 2 | Ahmadpure ast | 698 |
| 3 | Akhorakhatak | 923 |
| 4 | Arifwala | 446 |
| 5 | Attock | 597 |
| 6 | Badin | 227 |
| 7 | Bhawalnagar | 631 |
| 8 | Bahawalpur | 621 |
| 9 | Batkhela | 9323 |

- ASP Program to Display Selected Data from Table:

```
<html>
<head>
<title>oracle test page </title>
</head>
<body background="BUGTOWNBKG.GIF">
<center><h1>

<%
set objconn= server.createobject("ADODB.connection")
objconn.open "dsn=alvi;uid=;pwd=;"
city = Request.Form("D1")
set objrs=objconn.execute("select Imp_Places, Imp_Numbers from impnumbers where
City_Name= '"&city&"'")
%>

</center><font
color="#FFFFCC">         &nbs
p;      
<marquee width="417" height="36" align="middle" style="color: #005E5E"
bgcolor="#FFFFCC">IMPORTANT
TEL NUMBERS</marquee>
</font></h1>
<form method="POST" action="test.asp">
  <p align="left">CITY  : <select size="1" name="D1" style="background-color:
#FFEAEB">
  <option>SELECT</option>
  <option>Lahore</option>
  <option>Karachi</option>
  <option>Islamabad</option>
```

```
</select><input type="submit" value="GO" name="B1" style="background-color:
#DFF4DD; font-family: Times New Roman; font-size: 14pt; color: #000080; font-weight:
bold; border-style: solid; border-color:
#408080">           
; 
  <input type="text" name="T1" value="<%=city%>" size="16" style="background-color:
#FFE6E6">
<center>
      </p>
</form>

<%
response.write"<table align=center border=2 cellpadding=6 bordercolor=#800080
background=COLOR27.GIF>"
response.write"<tr>"
for i=0 to objrs.fields.count-1
response.write"<td><b>" & objrs(i).name & "</b></td>"
next
response.write "</tr>"
do while not objrs.EOF
response.write"<tr>"

for i=0 to objrs.fields.count-1
response.write"<td>" & objrs(i) & "</td>"
next

response.write "</tr>"
objrs.movenext
loop
response.write"</table>"
objrs.close
objconn.close
%>
```

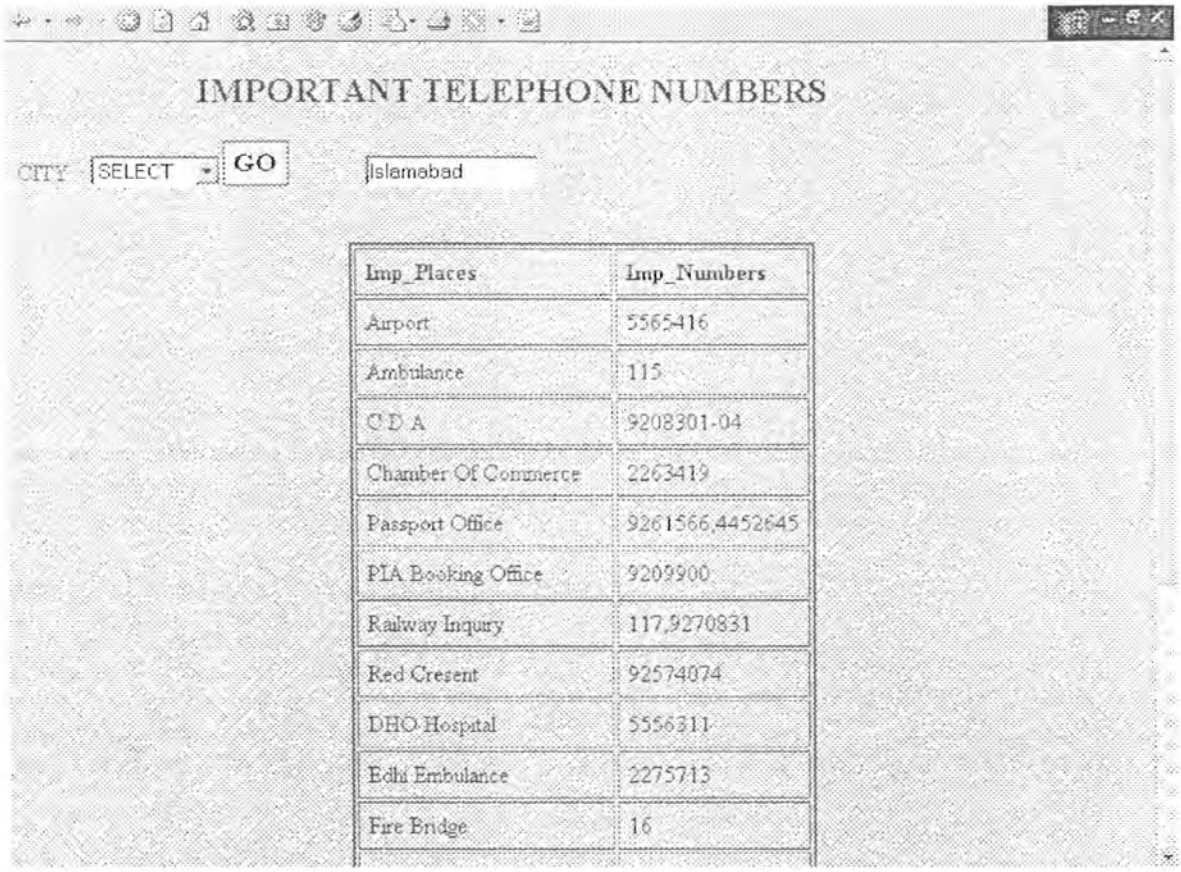
</body>

</html>

## Format Of Database:

Table Name: impnumbers

| *Field Name* | *Data Type* |
|---|---|
| *S_No* | *Text* |
| *City_Name* | *Text* |
| *Imp_Places* | *Text* |
| *Imp_Numbers* | *Text* |



# IMPORTANT TELEPHONE NUMBERS

CITY SELECT GO Islamabad

| Imp_Places | Imp_Numbers |
|---|---|
| Airport | 5565416 |
| Ambulance | 115 |
| C D A | 9208301-04 |
| Chamber Of Commerce | 2263419 |
| Passport Office | 9261566,4452645 |
| PIA Booking Office | 9209900 |
| Railway Inquiry | 117,9270831 |
| Red Cresent | 92574074 |
| DHO Hospital | 5556311 |
| Edhi Embulance | 2275713 |
| Fire Bridge | 16 |

# CHAPTER # 9

# SWISH 2.0

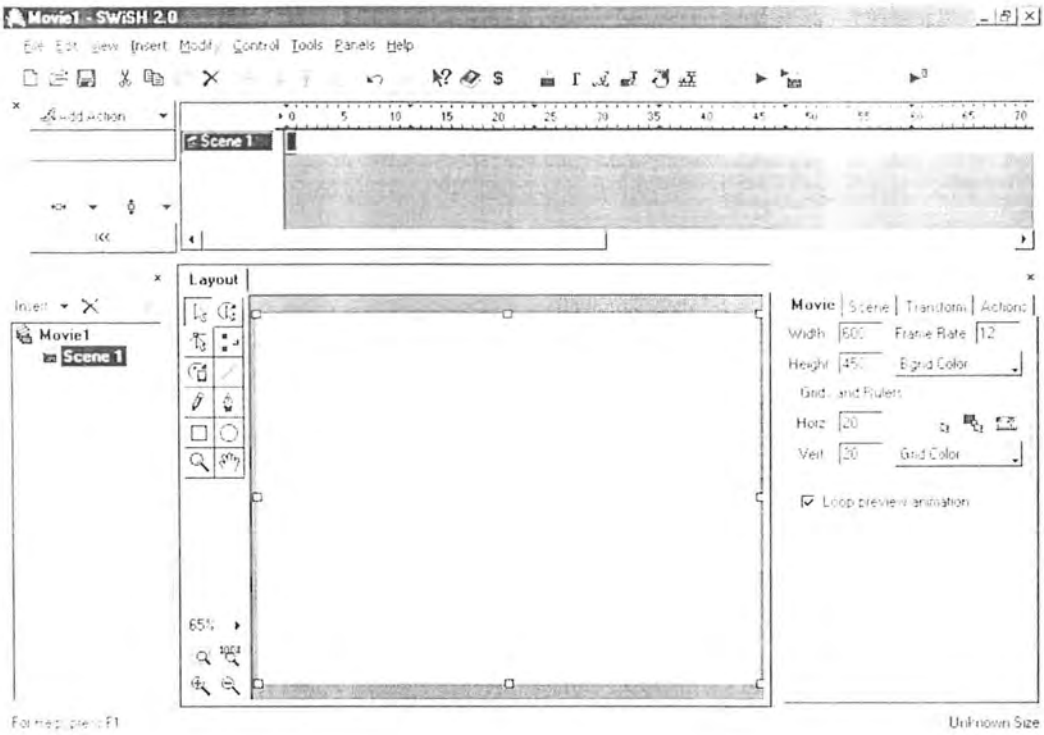
# SWISH (2.0)

## Introduction:

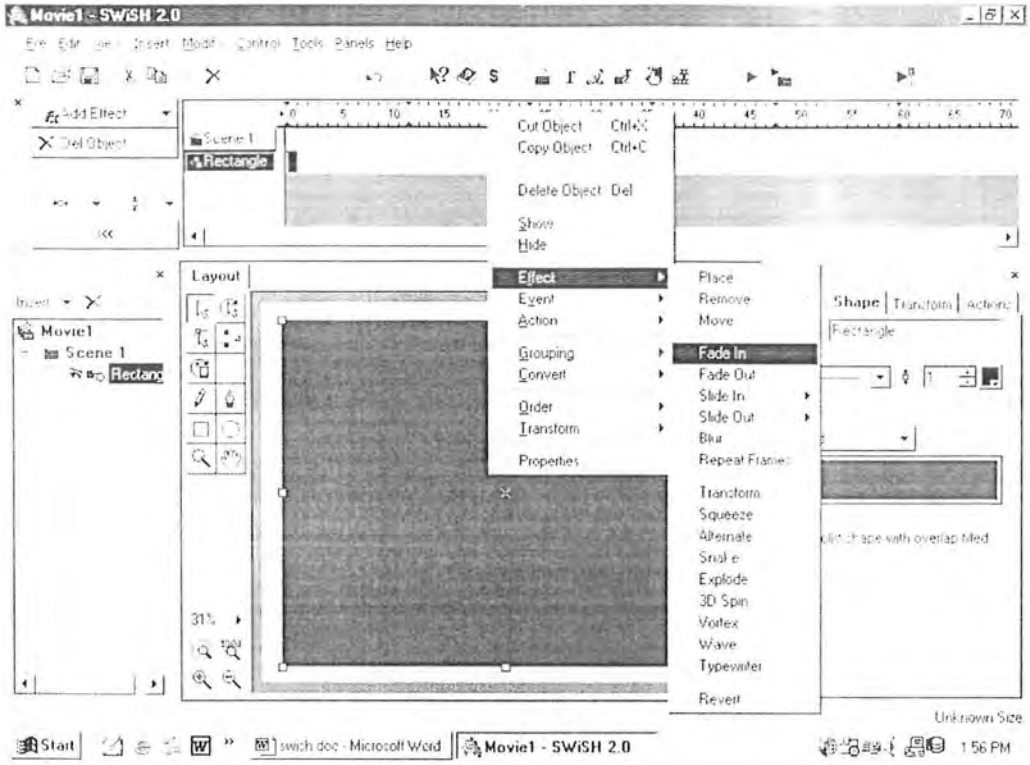
SWiSH is the quick, easy and affordable way to create Flash animations for your web site. With a few clicks of the mouse you can add a cool animated effect that will make your site stand out from the crowd. You can create shapes, text, buttons, sprites and motion paths. You can also include over 150 ready-to-use animated effects like Explode, Vortex, 3D Spin, and Wave. You can create your own effects, or make an interactive movie by adding actions to objects. SWiSH exports the same SWF format used by Macromedia Flash; so more than 97% of web surfers can see your animation without downloading a plug-in. You can preview your animation inside SWiSH without launching a browser or external player, and 'live-editing' lets you make changes while the animation is playing. SWiSH creates all the files you need to upload to your web server, or you can generate the HTML code to paste into an existing web page.

## Steps to create a Swish Movie:

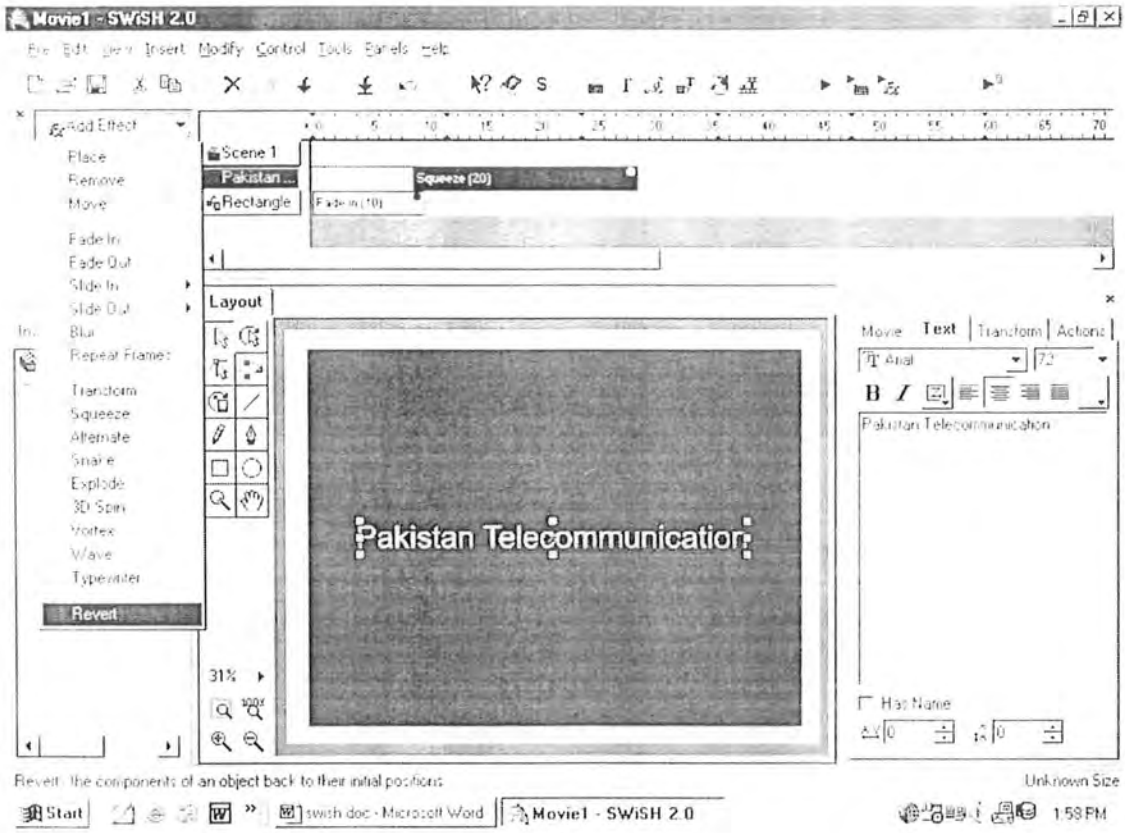
1. First of all while working in swish we decide the basic structure of our swf image. Then according to that we insert a scene and decide a background color. The canvas will look like this.



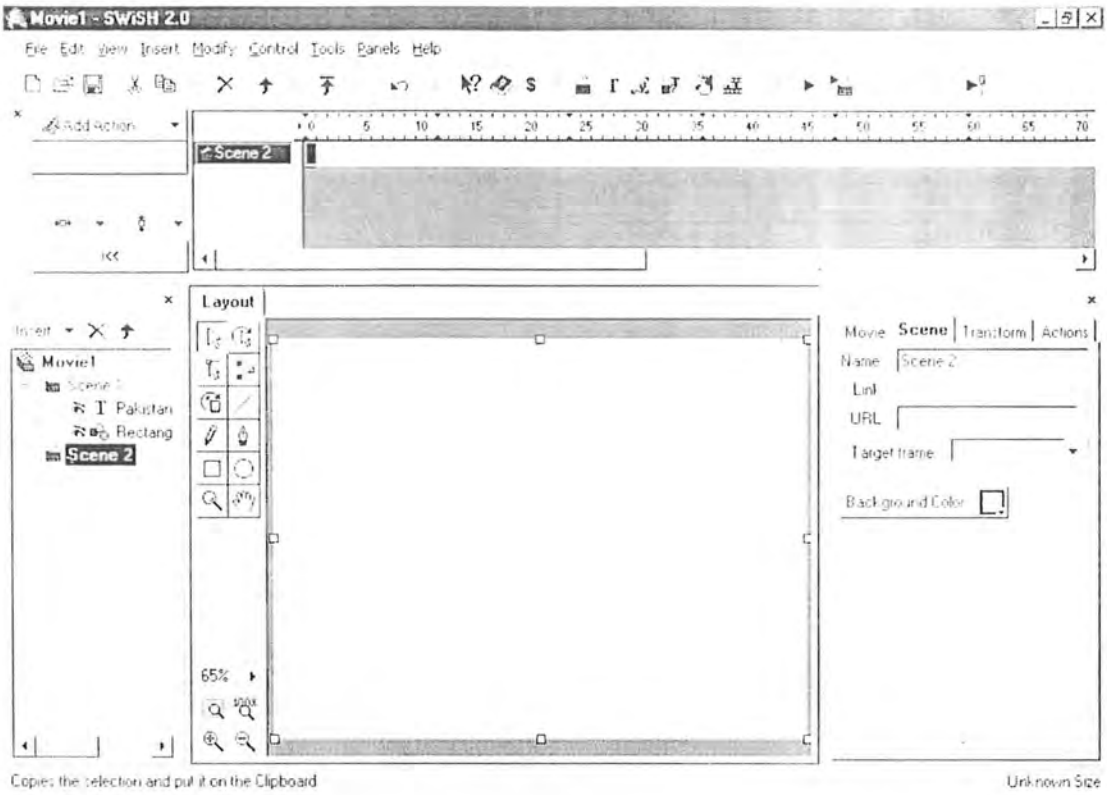

2. After applying background we select rectangle and give effect to it like Fade In.



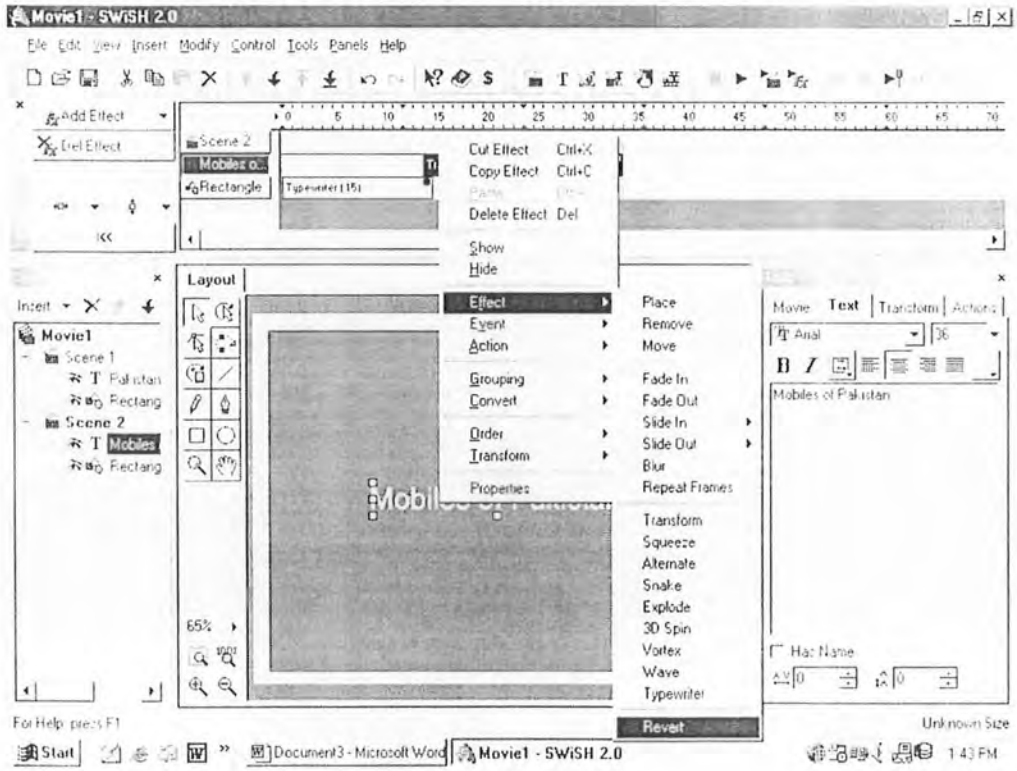

3. After this, we selected a text box from the insert menu. We write text and apply the required font size and font color from Text Toolbar and apply effects to it like squeeze.



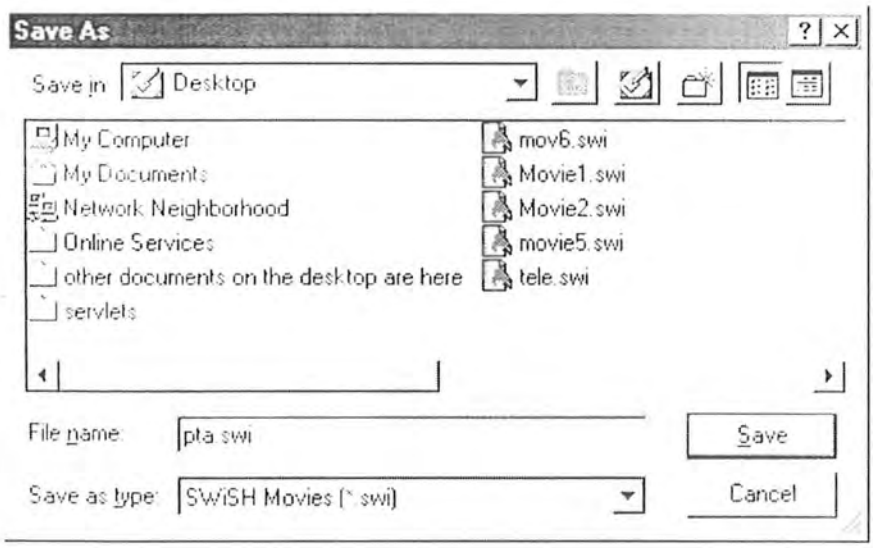

4. Then we select the second scene from insert menu and click on scene. Then second scene will look like this:



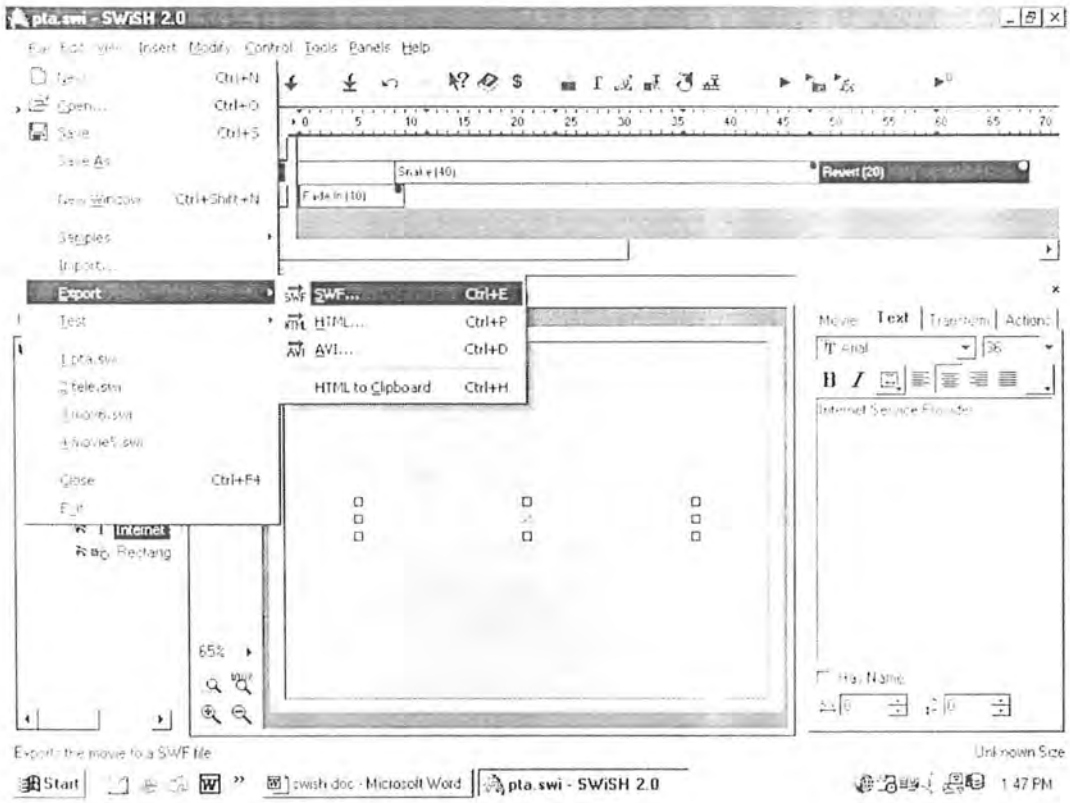

5. Repeat these steps to different scenes.



6. Now after completion of the Swish Movie we Save it with extension of .swi.




7. At last export it to SWF and save it with extension .swf.




## Inserting A Swish Movie to a Web page:

On the Web page, the place where you would like to insert that Swish movie, write a html code as

<embed src="pak.swf" width="130" height="75" align="right">

here the movie is saved by the name pak.swf and will be displayed at the right side of the web page.

# CHAPTER # 10

# CONCLUSION

# CONCLUSION

## Achievements:

- The project was to develop a web site giving complete information about Pakistan Telecommunication, which we have successfully achieved.

- There were no sites before which can give us information about Pakistan Telecommunication in detail. This web site provides all the necessary information about Pakistan Telecommunication.

- One aspect that is very important in web sites is the human computer interaction. We have tried to ease out all the stuff related to the user and try to make a site which user feels comfortable with and enjoy it.

- The site covered all the important aspects related to Pakistan Telecommunication. It is tried that all the important topics should be included.

- One aspect of the project is to connect a database in access to the web pages was successfully accomplished.

## Future Enhancement:

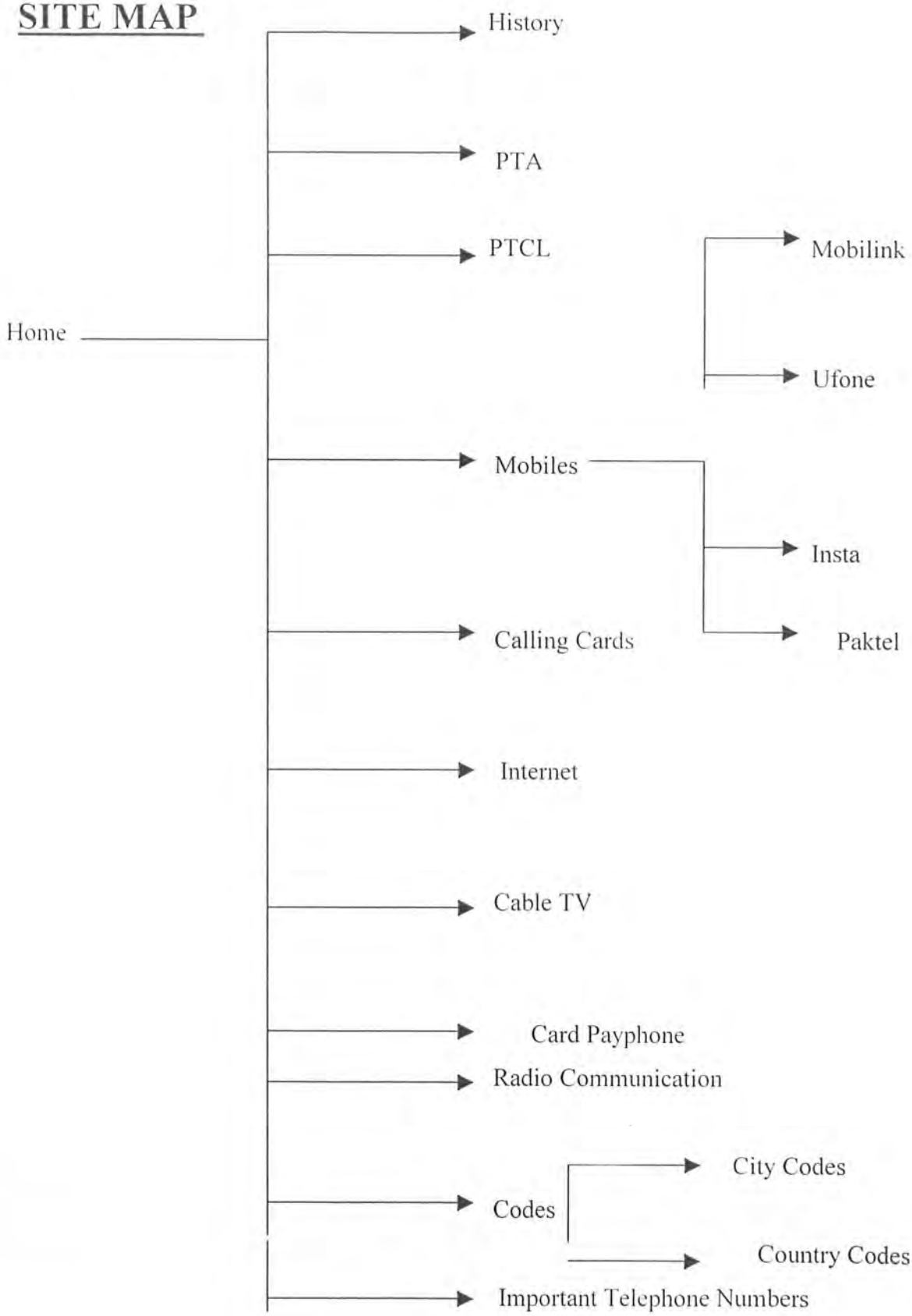
- Due to the limited time, many activities cannot be computerized. So in future some more things can be added to the site like Telephone Directory, etc.

- A form should be included in the web site to get suggestions from the people so that site can be enhanced as suggested by them.

- Due to limited resources, we tried our best to provide all the relevant information about Pakistan Telecommunication but much more can be added to the site in future.

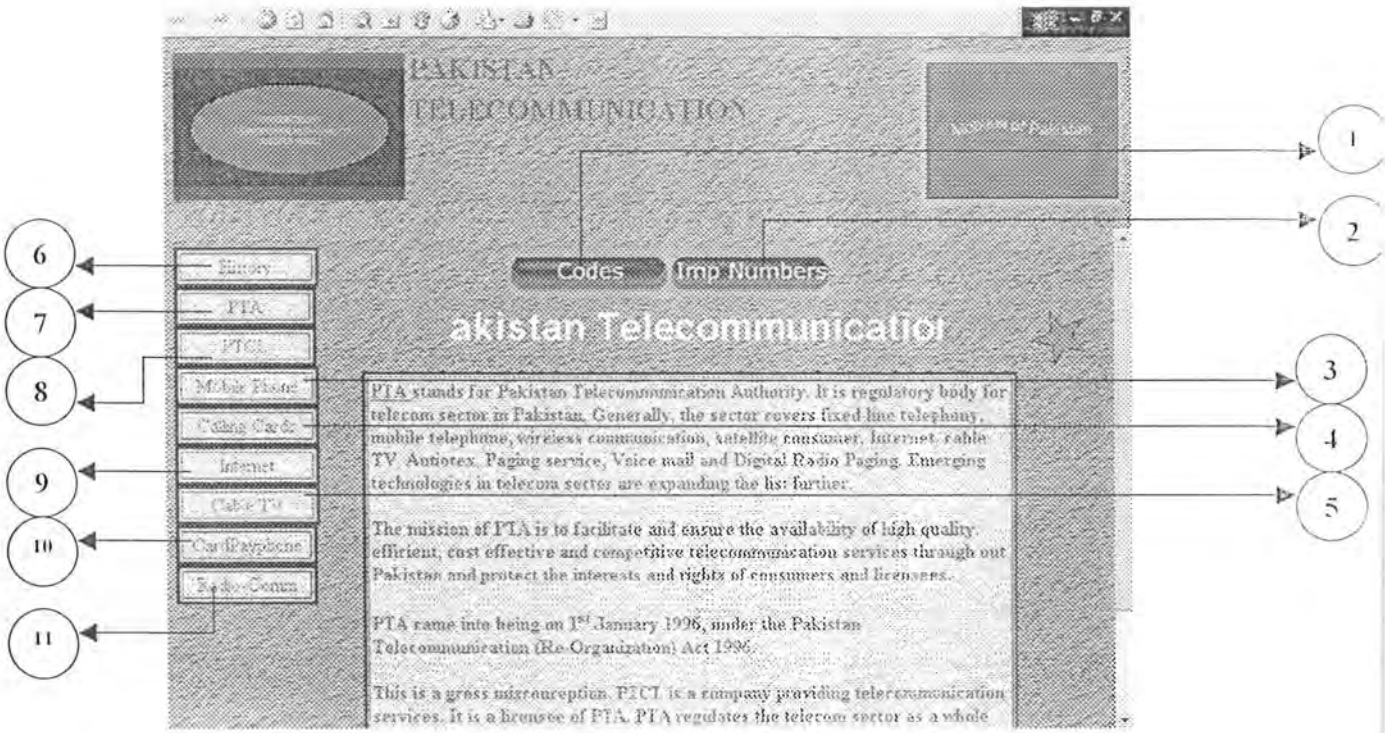
# SITE MAP

# SITE MAP


Home
History
PTA
PTCL
Mobilink
Ufone
Mobiles
Insta
Paktel
Calling Cards
Internet
Cable TV
Card Payphone
Radio Communication
Codes
City Codes
Country Codes
Important Telephone Numbers

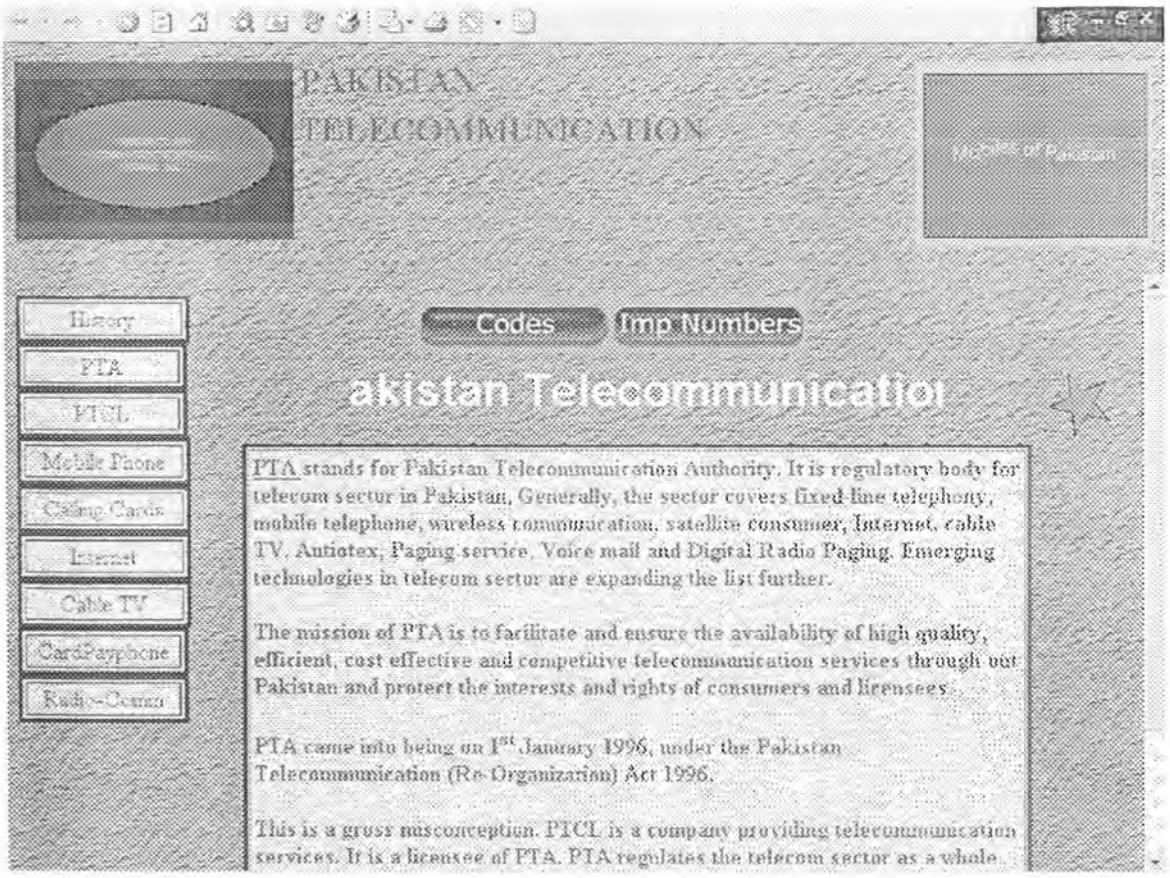
# USER GUIDE

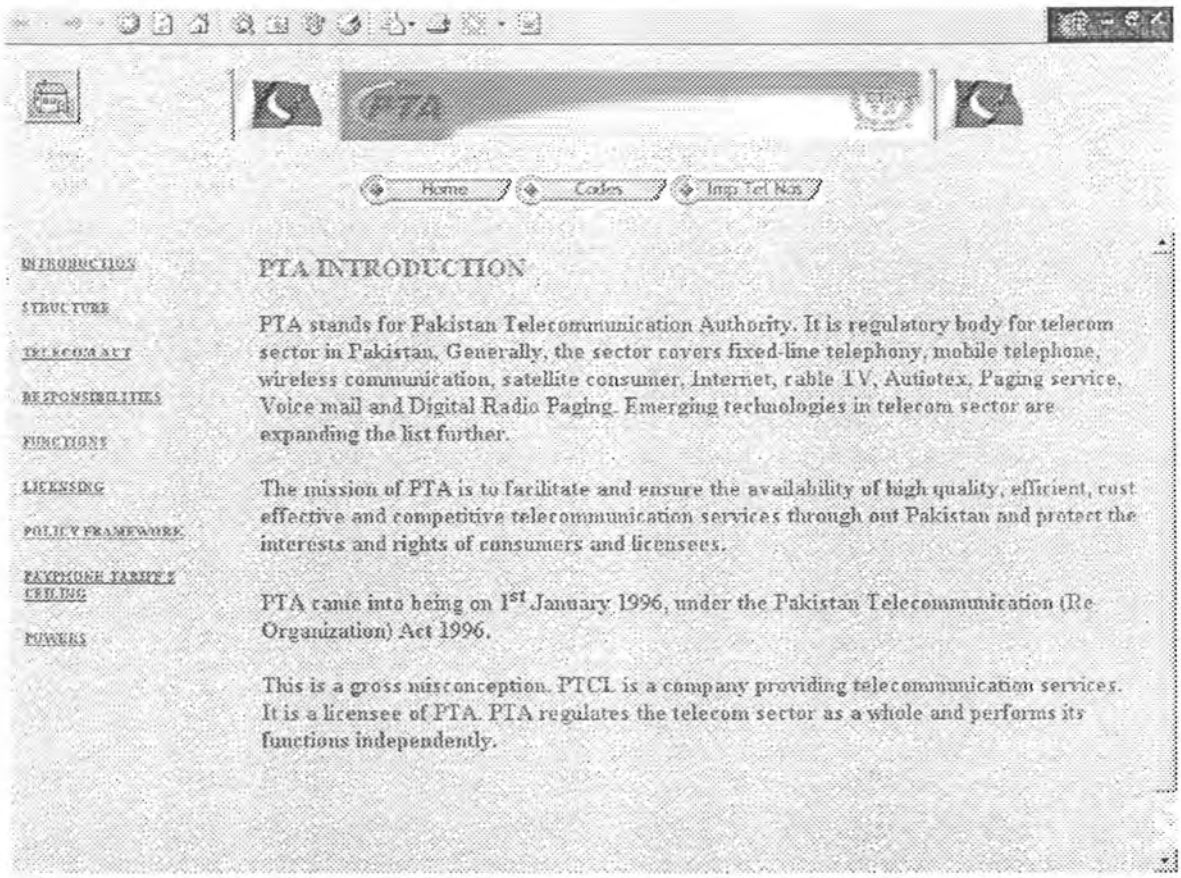
# USER GUIDE

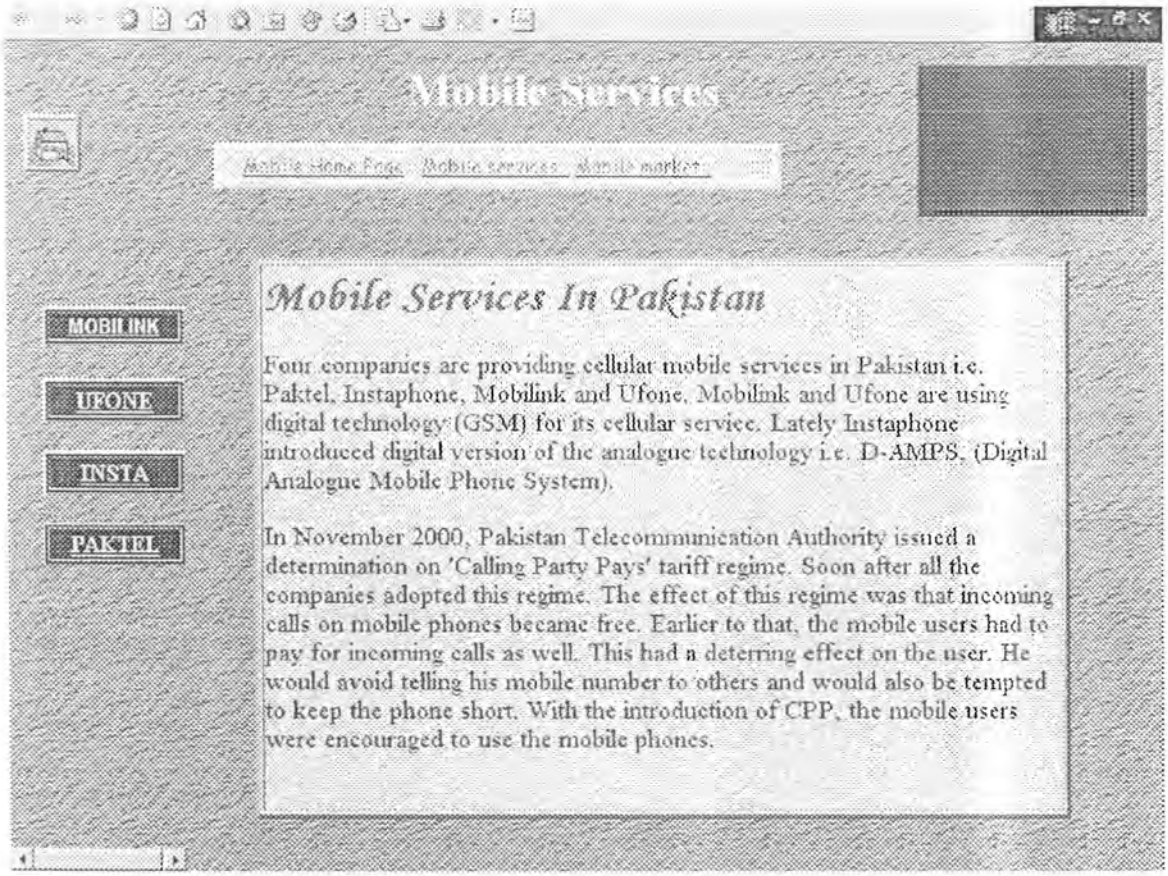

PAKISTAN
TELECOMMUNICATION
Codes
Imp Numbers
akistan Telecommunicatio
Mobile Phone
Calling Cards
Internet
Cable TV
PTA stands for Pakistan Telecommunication Authority. It is regulatory body for telecom sector in Pakistan. Generally, the sector covers fixed line telephony, mobile telephone, wireless communication, satellite consumer, Internet, cable TV, Autotex, Paging service, Voice mail and Digital Radio Paging. Emerging technologies in telecom sector are expanding the his: further.
The mission of PTA is to facilitate and ensure the availability of high quality, effirient, cost effective and competitive telecommunication services through out Pakistan and protect the interests and rights of consumers and licensees.
PTA came into being on 1st January 1996, under the Pakistan Telecommunication (Re-Organization) Act 1996.
This is a gross misreureption. PTCL is a company providing telecommunication services. It is a licensee of PTA. PTA regulates the telecom sector as a whole
1
2
3
4
5
6
7
8
9
10
11

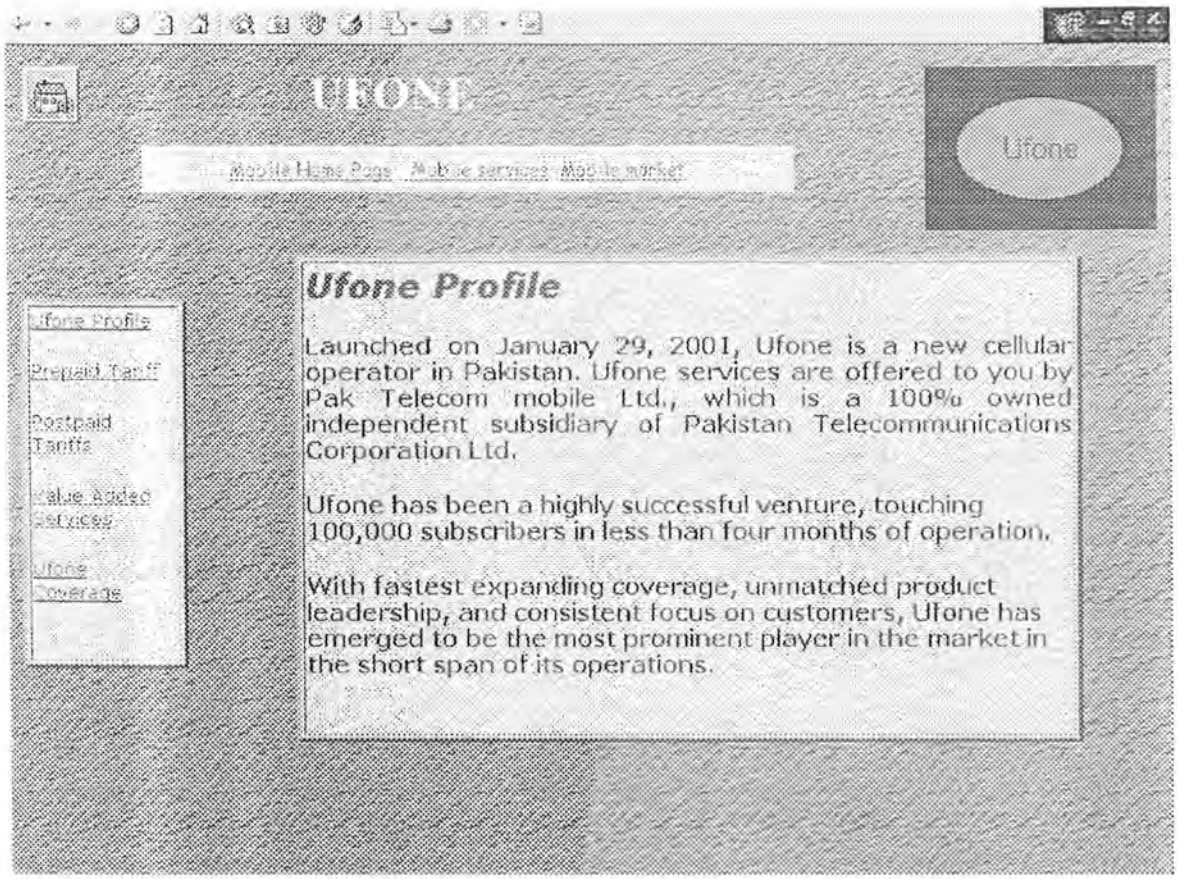
1. Click to know the city codes of Pakistan and country codes of the world.
2. Click to know the important telephone numbers of different cities of Pakistan.
3. Click to know the details of mobile services in Pakistan.
4. Click for the detail of Calling Cards available in Pakistan.
5. Click to know something about Cable TV.
6. Click for the History of Pakistan Telecommunication.
7. Click to know something about PTA.
8. Click to know about PTCL.
9. Click to know about Internet services in Pakistan.
10. Click to know about Card Payphones in Pakistan.
11. Click here to know about Radio-Communication in Pakistan.

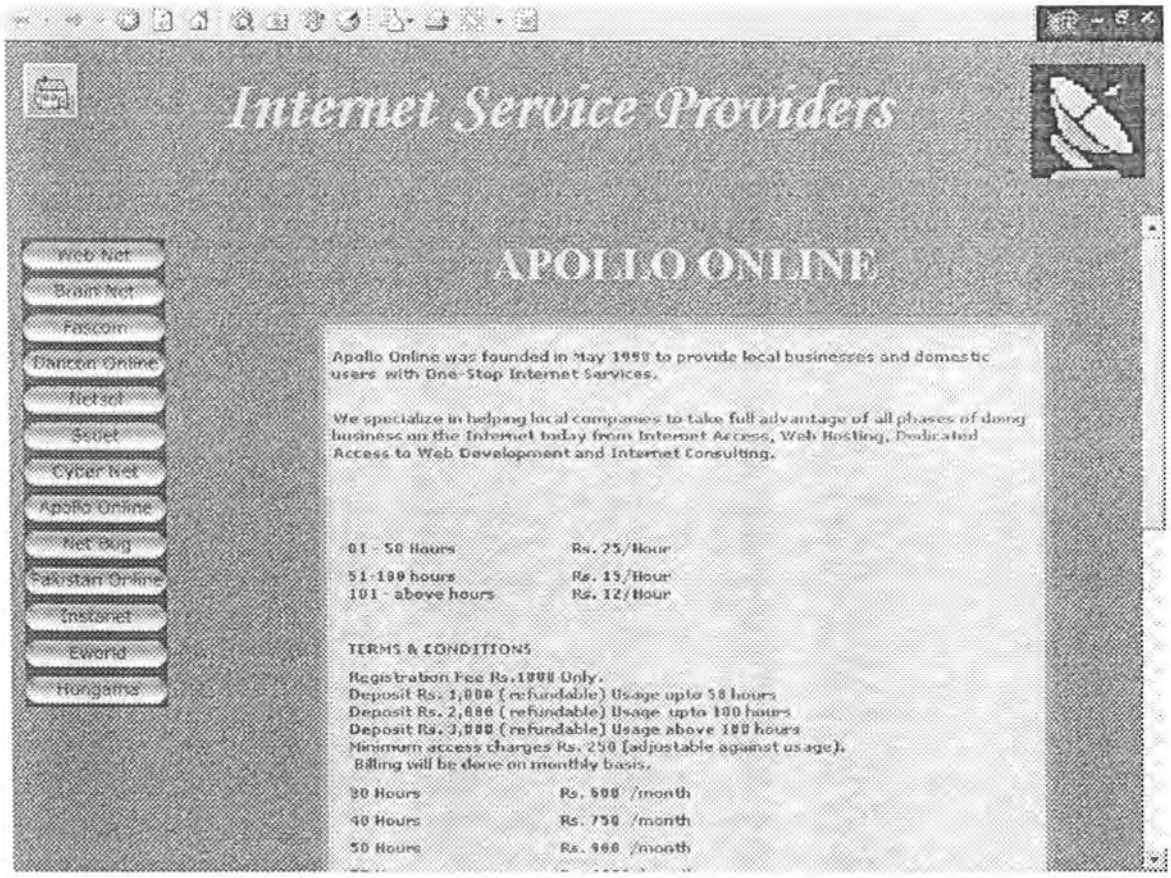
# TOUR OF THE SITE


PAKISTAN
TELECOMMUNICATION
History
PTA
PTCL
Mobile Phone
Calling Cards
Internet
Cable TV
CardPayphone
Codes
Imp Numbers
akistan Telecommunicatio
PTA stands for Pakistan Telecommunication Authority. It is regulatory body for telecom sector in Pakistan. Generally, the sector covers fixed-line telephony, mobile telephone, wireless communication, satellite consumer, Internet, cable TV, Antiotex, Paging service, Voice mail and Digital Radio Paging. Emerging technologies in telecom sector are expanding the list further.
The mission of PTA is to facilitate and ensure the availability of high quality, efficient, cost effective and competitive telecommunication services through out Pakistan and protect the interests and rights of consumers and licensees.
PTA came into being on 1st January 1996, under the Pakistan Telecommunication (Re-Organization) Act 1996.
This is a gross misconception. PTCL is a company providing telecommunication services. It is a licensee of PTA. PTA regulates the telecom sector as a whole.


PTA
Home
Codes
Imp Tel Nos
INTRODUCTION
STRUCTURE
TELECOM ACT
RESPONSIBILITIES
FUNCTIONS
LICENSING
POLICY FRAMEWORK
PAYPHONE TARIFF CEILING
POWERS
PTA INTRODUCTION
PTA stands for Pakistan Telecommunication Authority. It is regulatory body for telecom sector in Pakistan. Generally, the sector covers fixed-line telephony, mobile telephone, wireless communication, satellite consumer, Internet, rable TV, Autiotex, Paging service, Voice mail and Digital Radio Paging. Emerging technologies in telecom sector are expanding the list further.
The mission of PTA is to facilitate and ensure the availability of high quality, efficient, cost effective and competitive telecommunication services through out Pakistan and protect the interests and rights of consumers and licensees.
PTA came into being on 1st January 1996, under the Pakistan Telecommunication (Re Organization) Act 1996.
This is a gross misconception. PTCL is a company providing telecommunication services. It is a licensee of PTA. PTA regulates the telecom sector as a whole and performs its functions independently.


Mobile Services
Mobile Home Page
Mobile services
Mobile market
MOBILINK
UFONE
INSTA
PAKTEL
Mobile Services In Pakistan
Four companies are providing cellular mobile services in Pakistan i.e. Paktel, Instaphone, Mobilink and Ufone. Mobilink and Ufone are using digital technology (GSM) for its cellular service. Lately Instaphone introduced digital version of the analogue technology i.e. D-AMPS, (Digital Analogue Mobile Phone System).
In November 2000, Pakistan Telecommunication Authority issued a determination on 'Calling Party Pays' tariff regime. Soon after all the companies adopted this regime. The effect of this regime was that incoming calls on mobile phones became free. Earlier to that, the mobile users had to pay for incoming calls as well. This had a deterring effect on the user. He would avoid telling his mobile number to others and would also be tempted to keep the phone short. With the introduction of CPP, the mobile users were encouraged to use the mobile phones.


UFONE
Mobile Home Page
Mobile services
Mobile market
Ufone
Ufone Profile
Prepaid Tariff
Postpaid Tariffs
Value Added Services
Ufone Coverage
Ufone Profile
Launched on January 29, 2001, Ufone is a new cellular operator in Pakistan. Ufone services are offered to you by Pak Telecom mobile Ltd., which is a 100% owned independent subsidiary of Pakistan Telecommunications Corporation Ltd.
Ufone has been a highly successful venture, touching 100,000 subscribers in less than four months of operation.
With fastest expanding coverage, unmatched product leadership, and consistent focus on customers, Ufone has emerged to be the most prominent player in the market in the short span of its operations.


Internet Service Providers
Web Net
Brain Net
Pascom
Pakcom Online
Netsel
Scdet
Cyber Net
Apollo Online
Net Dig
Pakistan Online
Insternet
Everid
Hungama
APOLLO ONLINE
Apollo Online was founded in May 1998 to provide local businesses and domestic users with One-Stop Internet Services.
We specialize in helping local companies to take full advantage of all phases of doing business on the Internet today from Internet Access, Web Hosting, Dedicated Access to Web Development and Internet Consulting.
01 - 50 Hours Rs. 25/Hour
51-100 hours Rs. 15/Hour
101 - above hours Rs. 12/Hour
TERMS & CONDITIONS
Registration Fee Rs.1000 Only.
Deposit Rs. 1,000 (refundable) Usage upto 50 hours
Deposit Rs. 2,000 (refundable) Usage upto 100 hours
Deposit Rs. 3,000 (refundable) Usage above 100 hours
Minimum access charges Rs. 250 (adjustable against usage).
Billing will be done on monthly basis.
30 Hours Rs. 600 /month
40 Hours Rs. 750 /month
50 Hours Rs. 900 /month

# CALLING CARDS

Calling Cards Home Page | World Call | Call Point | Callmate | PTCL

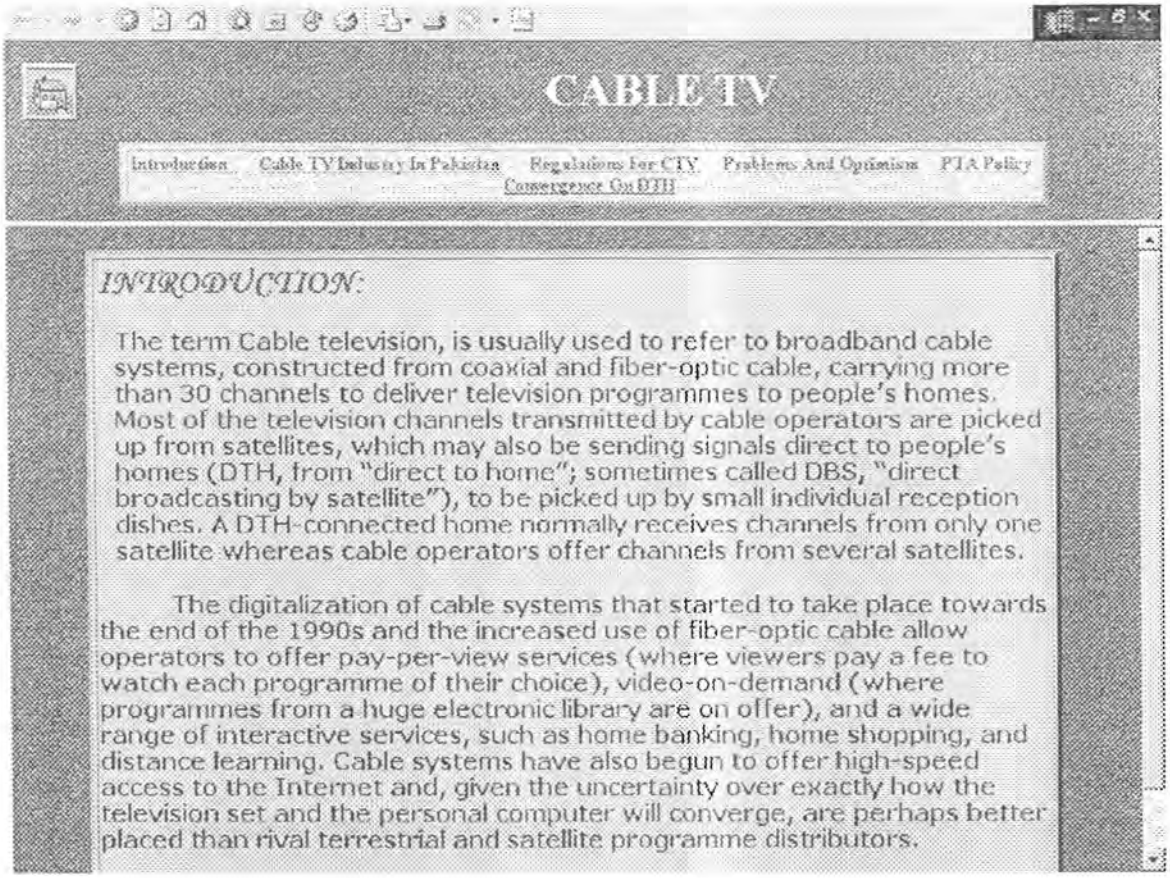
## WorldCall Calling Cards

WorldCALL undertook another pioneering effort when it became the first company to introduceinternational prepaid calling card service in Pakistan.

WPL launched its operations under the service brand name Hello ® in March 1999 and in a short span of time gained significant market share. The service was started in association with Unitel Communications, UK. Unitel is the turnkey technology supplier and also initially responsible for maintenance of the IN platforms.

Three IN platforms have been installed in Karachi, Lahore and Islamabad and the combined capacity now extends to 2000 ports. Domestic long distance dialing service feature was added on the same card in year 2000. Out of four service providers, including the state run monopoly Pakistan Telecommunications Company Limited (PTCL), WorldCALL holds 30% market share.

WorldCALL Phonecards Ltd is offering prepaid calling card services under an operation and maintenance (O&M) contract with PTCL. It offers the same rates as the PTCL service. However WPL's efficient service is supported by adequate capacity, state-of-the-art technology, customer support and a 24 hour customer helpline. Card distribution and availability to retail outlets is ensured by highly skilled sales team of WorldCALL Communications Ltd.


CABLE TV
Introduction
Cable TV Industry In Pakistan
Regulations for CTV
Problems And Optimism
PTA Policy
Convergence On DTH
INTRODUCTION:
The term Cable television, is usually used to refer to broadband cable systems, constructed from coaxial and fiber-optic cable, carrying more than 30 channels to deliver television programmes to people's homes. Most of the television channels transmitted by cable operators are picked up from satellites, which may also be sending signals direct to people's homes (DTH, from "direct to home"; sometimes called DBS, "direct broadcasting by satellite"), to be picked up by small individual reception dishes. A DTH-connected home normally receives channels from only one satellite whereas cable operators offer channels from several satellites.
The digitalization of cable systems that started to take place towards the end of the 1990s and the increased use of fiber-optic cable allow operators to offer pay-per-view services (where viewers pay a fee to watch each programme of their choice), video-on-demand (where programmes from a huge electronic library are on offer), and a wide range of interactive services, such as home banking, home shopping, and distance learning. Cable systems have also begun to offer high-speed access to the Internet and, given the uncertainty over exactly how the television set and the personal computer will converge, are perhaps better placed than rival terrestrial and satellite programme distributors.