Near East University





Department of Computer Engineering

Graduate Project

Com 400

Web Design and Development Tools

Student: Wael Fadel Hijazi

Supervisor: Miss Besime Erin

June 200

ACKNOWLEDGMENT

First of all I would like to dedicate this project to his almighty Allah whom enabled and supported me to complete this project.

In addition I want to thank all those who stood by me; my kindhearted mother, uncle Abu Osama and his family for their deep support and patience although my extended absence, uncle Abu Ala'a who always encouraged me and all my family members. I hope to be with them all in the near future.

A big thanks to my supervisor, Miss Besime Erin for her precious time and effort instructing me throughout this project.

Finally all my thanks to my friends; Raid Hamada, Mahmoud Ghanayem, Murat Kucuk, Tariq Elmasharawi.

Wael F.A Hijazi

988 .

ABSTRACT

This project takes a profound and explanatory look into the field of web design and development tools. This field is related to computer software and it needs more than software programs to work down it.

To start web design, designer must be professional in graphics field and HTML Editors alike.

According to this prefix, before introducing the this project and describe it, there is a quick chapters to describe and explain some of the basic concepts of creating and designing a web site like PhotoShop software, FrontPage, HTML Editors and Java.

Table of Contents

Acknowledgement
Abstract
Chapter 1: Introduction
Chapter 2: Visual Web Editor for Windows
2.1 FrontPage 98
2.2 How to Start FrontPage 98
2.3 How to Create A site
2.4 How to Create A Directory
2.5 Text Size, Color and Font
2.5.1 Changing Text Size
2.5.2 Change Text Color
2.5.3 Change Text Font
2.6 How to Add A Link
2.7 How to Add an Image
2.8 How to Add A Background
2.9 Creating A Table
2.9.1 Editing Table Properties
2.9.2 Editing Cell Properties
Chapter 3: HyperText Markup Language
3.1 What is HTML?
3.11 What is not HTML?
3.1.2 Why not use a program to make web pages?
3.2 Working with HTML Documents
3.2.1 How to name HTML Document ?
3.2.2 Why should save it with .HTML or .HTM

3.2.3 Opening the HTML Document into Browser
3.3 Tags
3.3.1 What is Tag?17
3.3.2 Tag Format
3.3.3 Tag What it Does?
3.3.4 Web Page Format
3.4 Attributes
3.4.1 What an Attribute?
3.4.2 Changing size of Text
3.4.3 Centering Text
3.4.4 Aligning Text
3.5 Making Links
3.5.1 Linking to other WebPages
3.5.2 Creating E Mail Link
3.6 Adding Images
3.7 Aligning Images
3.8 Question and Answers
Chapter 4: JAVA
4.1 Coding By Java25
4.2 why Java?
4.3 How to Start Coding By Java?
4.4 Object Oriented Programming
4.5 Data
4.6 Functions
4.7 Operators
4.8 Expressions
4.9 Interface
4.10 Declaring Classes

Chapter 5: Graphics by PhotoShop	36
5.1 The Layers Palette	36
5.1.1 Opacity	36
5.1.2 Preserve Transparency	36
5.1.3 Why use Layers?	37
5.2 The toolbar	37
5.3 Chrome Text	41
5.4 Glowing Text	42
5.4.1 How to Glow Text	42
5.4.2 How to break an Image	43
5.5 Fading Images	44
4.6 Animated Motion	46
4.7. Rounded Corners	48

Chapter 6: Online Applications	50
6.1 Computing Models	50
6.1.1 User Benefits of Online Application	51
6.1.2 Bad things	52
6.2 Online Enterprise Applications	53
6.3 Online End-User Applications	55
6.4 ASPs or ISP	57
6.5 Web Developer	58

Chapter 7: NEU Students Online Active Website	60
7.1 Overview	60
7.2 Index.html File	62
7.2.1 HTML Code	62
7.3 Main1.html File	65

7.3.1 Welcome Interface	65
7.3.2 Main1 File HTML Code	65
7.3.2 Main2.html File	68
7.4.1 E Mail Service Interface	68
7.4.2 Main2 File HTML Code	69
7.4.3 Login Menu	71
7.4.4 Signup Menu	71
7.4.5 Compose an E Mail	73
7.5 Main3.html File	74
7.5.1 Chat Interface	74
7.5.2 Chat Login Interface	74
7.5.3 Main3 File HTML Code	76
7.6 Main4.html File	78
7.6.1 Books Online Interface	78
7.6.2 Main4 File HTML Code	79
7.7 Main5.html File	85
7.7.1 Link and Search Interface	85
7.7.2 HTML Code	89
7.8 Main6.htm File	95
7.8.1 About Interface	95
7.8.2 About File HTML Code	95
7.9 HTML Files Flowchart	97
7.10 Files Interfaces Flowchart	98
Conclusion	99
References	100

Chapter 1

Introduction

I Choose this project to work in for many reasons, one of which my desire to do something worth for NEU university. Another reason was selfdeveloping project for my own especially in Web Design field.

It is hard to understand such a project if an introduction to basic concepts of web creation and design was not shown in previous.

These explanations I enhanced here leads not only to the understanding of my project, yet. Beyond that it helps beginner learn how to create, design and develop his own website.

Chapter 2

Visual Web Editor for Windows

2.1 FrontPage 98

FrontPage 98 is one of the best HTML development tools available on the market. This Chapter shows how to use FrontPage 98 or if the user have knowledge of the program, will refine his understanding.

Professional Web site publishing is easier than ever before with Microsoft FrontPage 98. This Web site creation and management software provides a quick, effective way to create and manage professional-quality Internet or Intranet sites--without programming! With innovative imaging tools and intelligent design assistance, Microsoft FrontPage 98 is ideal for new users and professional developers alike.

This is the Zeus of HTML Editors. FrontPage 98 can help you make your web site with that feel you want. Unlike bulky HTML editors, FrontPage 98 gives you the flexibility that is required when developing your web site.

2.2 How to start FrontPage 98

After installing FrontPage 98 to a Windows 95/98/NT system, User can start the program by clicking Start, moving up to Programs and clicking the Microsoft FrontPage icon in the Programs Menu.

Microsoft FrontPage

When the program is opened there are two windows:

• FrontPage Explorer: The FrontPage Explorer is used to manage you website. From here you can easily rename, move, edit, or add files to your FrontPage

2

web. User can also view the structure of his entire website through the FrontPage Explorer.

• FrontPage Editor: The FrontPage Editor is where all of the actual web page designing and creation takes place. From here user can add text, images, forms, scripting languages (Active X, Java, etc...), sound, and video.

2.3 How to Create A site

These simple steps show how easily to create the foundation of a web site. FrontPage thinks of a web site as a tree of web pages. User must setup the structure of the tree before creating the actual pages.

First, click on the Create a New FrontPage Web radio button in the Getting Started Dialog Box.

1 Sta	Open an Existing FrontPage We	Ь.,	
	<root web=""> <root web=""> <root web=""> <root web=""> <root web=""> <root web=""> <root web=""></root></root></root></root></root></root></root>		Aore Webs
	Create a New FrontPage Web		
States and Street	s op n fast web		
	and the second states and the states of the	and the Ball of the State of the State of the	

Figure 2.1

Click the OK button (Fig 2.1).

A New FrontPage Web Dialog Box will open (Fig 2.2)

From here select the kind of FrontPage web to create. In most cases this will be the One Page Web or the Personal Web. Type in the name of your FrontPage website. Note: This is only a reference name. What you enter here will have no effect on the creation of the web itself.

A Cha	ose the kind of FrontPage web to create:
	Ine Page Web
1	C Import an Existing Web
	From Wizard or Template:
	Customer Support Web Discussion Web Wizard Empty Web Learning FrontPage Personal Web Project Web
Che	a tille for your FrontPage web
1	School Name
	http://www.school.edu/schoolname
	Change.

Figure 2.2

Click on the Change button. (fig 2.2).

the Change Location Dialog Box enter the location you would like your web site to be created at.

Note: The name you enter here will be the root directory of your web site.

For example if you enter www.schoolname.edu/schoolname that would be the address you would have to type in on the browser to visit your site. Click on the OK button.

Note: You may be asked to enter in your name and password. Your server administrator should provide you with one. Now your setup to create the pages.

Your server should be setup and ready to go! Excited? You are about to create your first web page using FrontPage 98.

Now you're set to create your first web page. Start by choosing the directory that you want the new page to be located in. Directories are important to keep your documents in order. In the next Chapter, I show how to make more than just your root directory. Click on New Page icon in the Toolbar.

New Page

This will create a new web page in the directory that you currently have selected. To edit your web page you need only double-click on the page and it will open the web page in the FrontPage Editor. After creating a new page, you have several options. In further Chapters, I explain how to change the text, add links to other web pages, & add images. Now that you know how to start making web pages, you probably know that web pages are made in HTML. In the next section, you will learn how to create a directory. Directories help you keep your web site organized.

2.4 How to Create A Directory

Directories are important to web development. As your web site grows, it will be more and more difficult to keep track of where all your files are. By making directories, it is much easier to organize your site. You can make a directory by simply selecting File from the Menu Bar, then selecting New from the File Menu and finally selecting Folder from the New Menu (fig.2.3).

<u>ha</u> the	Eage	Ctrl+N
Open FrontPage Web	<u>T</u> ask	Clif+T
Close FrontPage Web		

Figure 2.3

A new folder will appear in the content section of the FrontPage Explorer (fig 2.4).

Eordents of Images			
Name	Title	Size	Туре
changelocation.gif	images/changelocation.gif	2KB	gif
fptutorial.GIF	images/fptutorial.GIF	4KB	GIF
freegraphics.gif	images/freegraphics.gif	4KB	gif
freegraphicsarrow.gif	images/freegraphicsarrow.gif	4KB	gif
frontpage.gif	images/frontpage.gif	416	gif
gettignstarted.GIF	images/gettignstarted.GIF	6KB	GIF
gettingstarted.gif	images/gettingstarted.gif	2KB	gif 📰
leadin.gif	images/leadin.gif	22KB	gif
namenpswd.gif	images/namenpswd.gif	ЗКВ	gif
🗱 newfpweb.gif	images/newfpweb.gif	8KB	gif
programs.gif	images/programs.gif	373	gif
🛞 resume.gif	images/resume.gif	3KB	gif
Resumearrow.gif	images/Resumearrow.gif	ЗКВ	gif
samplework.gif	images/samplework.gif	4KB	gif
sampleworkarrow.gif	images/sampleworkarrow.gif	4KB	gif
startbutton.GIF	images/startbutton.GIF	301	GIF
🗱 title.gif	images/title.gif	5KB	gif
title_aboutme.gif	images/title_aboutme.gif	4KB	gif
title_freegraphics.gif	images/title_freegraphics.gif	5KB	gif
title_resume.gif	images/title_resume.gif	ЗКВ	gif
title_samplework.gif	images/title_samplework.gif	5KB	gif
newpage.gif	images/newpage.gif	249	gif
Mere Folder			folder
		** 1990/1990/1999	

Figure 2.4

Its that simple. Now that we have the site setup and organized, you will need to learn how to actually create the web pages. First, you will learn how to edit your text.

2.5 Text Size, Color & Font

This section, describes how to change the text size, color & font. There are a number of different ways you can accomplish these tasks. However, I will show you what I believe is the easiest way.

2.5.1 Changing Text Size

Select the text whose size you would like changed.

Select either the Increase Text Size or Decrease Text Size buttons. You can continue to keep selecting these until you achieve your desired text size.



2.5.2 Change Text Color

Select the text whose color you would like changed.

Select the Text Color Button



user can choose any of the basic colors (Fig. 2.5) or if you want to be really creative

user can click on the Define Custom Colors Button. From the Define Custom Colors (Fig. 2.6) add-on you can adjust the sliders you find the color that matches you need.

Once you have found the color you would like to use select the Add to Custom Colors button.

Select the custom color you just created.

Click the OK button.

A asic colors

X

Figure 2.5



Figure 2.6

Chapter 2 - Visual Web Editor for Windows



Figure 2.7

Select the text or image you would like to be a link to your e-mail address. Click on the Create or Edit Hyperlink Button. Then click on the Make a Hyperlink that sends e-mail button.

Type in your e-mail address and click OK. Its that easy. Now I will show how to add an image to your webpage.

2.7 How to Add an Image

Images are essential to your web site. They are the eye candies that will make your visitors happy with what they are seeing. This leads to the visitors coming back again and passing on the URL to others. FrontPage makes it simple to add images. It only requires a few clicks of the mouse. Start by clicking on the Insert Image button.

Then select the image you would like to insert into your web page.

Click OK and the image appear (Fig. 2.8.).



Figure 2.8

2.8 How to Add A Background

Backgrounds are beneficial to your site if they are used properly. Make sure the visitor can read the text on the background you selected. To add a background in FrontPage begins by opening the page that you would like to change the background on. Select File from the Toolbar. Then select Page Properties from the File Menu.

In the Page Properties Dialog Box click on the Background tab. You can select the background color from the background drop-down box, or you can choose a specific image by selecting the Browse button and then choosing the image you would like to use. Click the OK button (Fig. 2.9).

Chapter 2 - Visual Web Editor for Windows



Figure 2.9

That's the very basics of using FrontPage 98. In next section I describe more advanced topic such as tables. And this is easy to create using FrontPage 98. Because FrontPage 98 is such a complex program, we have not covered all topics. However, this introduction should be enough to get you started on a nice web

2.9 Creating A Table:

Click and hold the left mouse button down on the Insert Table Button (Fig. 2.10). Drag the mouse down and to the right to choose the size of the table you desire.

Release the mouse button and you have your table.

2.9.1 Editing Table Properties

Select any cell in the table you want to edit. Select Table from the Toolbox Menu.

Select Table Properties from the Table Menu. (Fig. 2.11)



A Table Properties Dialog Box will appear (Fig. 2.12).

Figure 2.10

2.9.2 Editing Cell Properties

- Select the cell that you would like to edit.
- Select Table from the Toolbox Menu.
- Select Cell Properties from the Table Menu.

A Cell Properties Dialog Box will appear (Fig. 2.13). From here you can change the alignment (vertical & horizontal), width and height, background and color, and spanning.

When you have finished making your changes click the OK button.

Table Rome Window, Help Draw Table. Insert Lable Insert Brown or Columna ... Insert Cell Insert Caption Celene Gelia Metro Metro Sent Cent Select Cell Select How Select Column Setent Table BOMBLIE BILLISE VE IV Bronkide Volumnis Evenig Ennien Techno Teche **Convert Table To Text** Labio Angenesi Cell Properties... Table Propertie:

Figure 2.11



(Figure 2.12)

Cell Frozenie :				23
Kayook Horgontal Alignment Vertical Alignment	Default	Mismun Size	E IN FIGHT	0K Cancel
THeader Eet 1	NoWrap	Specify Height	Criptfors Griptfors Griptfors	Apply Style
Custom Background Use Background Im Background Color	dge Default	Browse These	<u>ner 1</u>	
Euston Colors Border: Light Border: Dark Border:	Default Default Default	Cell Span Number of Bows Spanned: Number of Columns Spanned:		

(Figure 2.13)

Chapter 3

Hyper Text Markup Language

(HTML)

This chapter is to learn a new aspect of HTML or just to refine your understanding on a certain topic.

By the time you reach the last basic section, you will be able to make a web page that is customized to your own needs. It's not that difficult! Hundreds of thousands do it.

3.1 What is HTML?

It's what enriches plain-text information to be the colorful, interactive web pages that you see all across the web. Without HTML the term "surfing the web" would mean "digging through mounds of text".

Hyper	Text	Markup	Language
No longer does the reader have to read things in order. This enables them to skip around between sections. This "loss of order" is what makes HTML hyper.	This is what you will use to convey your ideas to the reader. It's good or English letters.	In a web page, HTML marks up the text. This is what HTML is all about.	No, you can't speak it like English (well, you probably could, but it might get ugly). It's the language used to create web pages.

3.1.1 What isn't HTML?

It's not a computer language. This can't be emphasized enough. If someone asks you what computer languages you know, don't answer "HTML.". Also, HTML isn't a difficult, cryptic computer language. It is a set of codes that enables rich types of content to be distributed across the World Wide Web.

3.1. 2 Why not use a program to make web pages?

With all those programs that let you create web pages without learning HTML, why learn the language anyway? Here are three important reasons why HTML is the best way to go. These such programs are:

- Windows 95/98/NT, Macintosh, or UNIX. Save your files with the .html extension. Name your files whatever you want (avoid the space, quotations, and any other weird characters) and add the suffix .html. Here are some legal examples of filenames: dogs.html, myfirstpage.html, htmlrules.html, a_really_long_but_legal_filename.html. I would suggest you stay away from capital letters to reduce confusion.
- Windows 3.x. You are limited to naming your files in the 8.3 format. For an HTML document this means that the filename must be eight letters long with the three letter extension, *.htm.* Here are some legal examples of filenames: *anything.htm, chris.htm, computer.htm.*

3. 2. 2 Why should save it with the .html or .htm extension?

All this extension mumbo-jumbo is based on a thing called file association. When you add the .html or .htm suffix to your document, your computer knows it's an HTML document. If you learn about graphics, movies, or sound, then you will see other, different file extensions.

3. 2. 3 Opening the HTML document into your browser

When you've got an HTML document saved onto your hard drive or floppy disk, follow these instructions to see how it looks in a browser.

- 1. Go to the File menu and select Open, Open File with Browser, Open Page, or anything very close.
- 2. Some browsers let you select the file right away, but on others you will have to click a *Browse*... or *Choose File*... button. When you get the dialogue box, find your HTML document (whatever you named it) and click the *Open* button. You may have to *OK* or *Open* out of another window of a newer browser.
- 3. The browser will do the rest and hopefully renders your HTML document so it appears as a web page.

<u>3.3 Tags</u>

Now that you've played with your text editor and tried saving as .html or .htm, we're ready to get down to writing. In the next section you will create you first web page.

3. 3. 1 What is a tag?

HTML is written in Plain English Text. It's simple, reads top-to-bottom, and reads left-to-right, but it isn't just regular English text. What makes it different? Tags. You will use tags to set text apart to be big, small, bold, underlined, italicized, and many more.

Tags are many like light switches in the way they act. If you want a sentence to be bold, you place the bold tag at the exact spot where you want the bold lettering to start (like turning on a light switch) and where you want it to stop (like turning off a light switch).

3. 3. 2 Tag Format

All tags begin with a less-then sign (<) and end with the greater-than sign (>). This will be the format for every future HTML tag. Between the <> is a code which depends on what you want to do. Learning HTML is about learning codes to perform what you want to happen. Here's an example. Let's say we want to write the word "Dog" in bold. The HTML code for bold is B.

Dog

Take a look at what is happening...

Dog

This is the beginning bold flag. It, like a light switch, turns on the bold and is called the opening tag. With the bold tag preceding it, this text is turned bold. This turns off the bold and is called the closing tag. If there were more text after this closing tag, it would be just normal text.

This entire thing is called an element. It consists of an opening tag, the affected text, and a closing tag.

More elements for you to use

Here is a table of the B element (used in the example above) and three new ones that can be used just like the bold element:

Code Effect Tag in Use How it looks B Bold Bold Bold TT Typewriter <TT>Typewriter</TT> Typewriter I Italics <I>Italics</I> Italics U Underlined <U>Underlined </U> Underlined

<U>Bold and Underlined</U> becomes Bold and Underlined <I>Italicized and Bold</I> becomes Italicized and Bold <I><U>Italicized and Underlined</U></I> becomes Italicized and Bold

Elements with just one tag

Unlike the four elements you have learned already, these elements have only a single tag. As you will see, they do not need a second tag...

3.3.3 Tag What It Does

<HR> This element puts a line across your page. It stands for Horizontal Reference.

The line directly above this text was created using <HR>

 $\langle BR \rangle$ This stops text on the current line and begins again the next line. A Break in other words. You will probably need this often, for when HTML is saved as text it does not save where you hit the ENTER key. This is how it works...

My favorite animal

is a cat.

The line break above where "My favorite animal" is broken from "is a cat." was caused by
.

<P> This is the equivalent of two
 tags. While
 just skips to the next line, <P> skips a line then starts again.

The whitespace you see just above this line of text was created from a <P> tag.

3. 3. 4 Web page format

Ready? You are about to create your first whole web page. We will be using elements you've learned already plus three more that you will learn below. You will see the following three elements on every page you make from now on.

All your pages will being with: <HTML>

this is how it looks in a web browser:

Wow! It's different! Remember that the <HR> element without an attribute looks like the following horizontal line:

So what happened? The added attribute, SIZE="6", modifies the element it's in, <**HR**>. This is the reason that the first horizontal line has a height of six. If you wanted to you could change the value of the SIZE attribute to something other than 6. Let's say we wanted a horizontal line with a height of 10. You would use: <**HR SIZE="10">**. It would appear like this in the browser...

4.2 Changing the size of text

To change font sizes you use the **** tag with the **SIZE** element. There are twelve (12) sizes for you to use: **-6 through -1, and +1 through +6**. Minus 6 is the smallest while +6 is the largest. Here's how they look on the page: (Fig. 3.1)

 $< FONT SIZE="-3">Size -3 \\ < FONT SIZE="-1">Size -1 \\ < FONT SIZE="+1">Size +1 \\ < FONT SIZE="+3">Size +3 \\ < FONT SIZE= +3$

Figure 3.1

See what's going on? The SIZE attribute modifies the FONT tag to change the text to the size you choose. Notice that the closing tag for an attributed element need not the actual attribute. For example, to close a tag, all you need to do is put (not or).

3. 4. 3 Centering Text

There are two ways to center text. They are:

<CENTER>This text is centered with the CENTER element </CENTER>

<DIV ALIGN="center">This text is centered with the DIV element and the ALIGN attribute</DIV>

I always use the **CENTER**> element to center text. I just wanted to show you the latter way to give you a little more practice with attributes. The **ALIGN** attribute tells **DIV**> that it should align to the center.

3. 4. 4 Aligning text to the right:

Here is how you can align text to the right side:

<DIV ALIGN="right">This is on the right</DIV>

Notice this **<DIV>** element can also be used to center text, as shown in the "Centering Text" section above. What enables you to align text to the right is the **ALIGN** attribute. When you set it to equal "right", then it goes to the right. If it's set to equal "center", it goes to the center. Simple, huh?

Now you have a very solid understanding of the format of HTML. From here on it'll be applying knowledge you have already learned. In other words "It's all downhill from here." Questions? Head onto the forums. Or...

3.5 Making Links

In this section you will learn how to link to other pages and your email address. These links are what make the HyperText Markup Language so hyper.

3. 5. 1Linking to other web pages

This is how you would link to the main www.neu.edu.tr page:

Go to NEU university

Chapter 3 – Hyper Text Markup Language (HTML)

A	HREF	http://www.neu.edu.tr	Go to NEU university	/A
This stands for Anchor. It is the element name for a link.	This stands for HyperText Reference. This is where the link will go when it's clicked.	This is the full address that the viewer will be sent to if the link is clicked.	What you want the user to click is what this is. This is what must be clicked for them to be sent to the link's destination.	The link element is ended.

Let's take a look at what's going on...

3. 5. 2 Creating a link so people can e-mail you

This will create a link to your e-mail address so anyone can e-mail you. Here's the format:

Click Here to E-mail Me

It's pretty similar to a link that goes to another page. The main difference is the full address is in a different format. You must use the "mailto:email-address" format to make it work.

Try clicking on it. It should pull up a window where you can easily e-mail me. Please do if you wish.

In the next section you get to learn how to put pictures on your web page. Ready?

3. 6 Adding Images

What makes HTML really fun is putting pictures on your web page. But how do you get those beautiful, intricate images into a simple HTML document? You don't. Images are separate files you include into your web page.

• Adding an image to your page

Here's the format ...

This is what it looks like through the web browser after replacing *image.gif* with neu.gif (some of my own work)...



IMG	SRC	image.gif
This stands for "image" and tells the browser that an image will go here.	This attribute stands for "source" and it tells the browser where to find the image.	This is the value of the SRC attribute and it tells the browser where the image is. The .gif extension tells the browser what type of image it is .

• How to Link An Image:

Now you know how to make a text link. You know how to place an image on your web page. Let's combine the two. We're going to make an image clickable, just like a link, so the viewer to click on it to get to another page. Here's the HTML for it:

It's just like a link, except you put an image instead of the text. Now, instead of clicking on text to get to *http://www.neu.edu.tr*, you click on an image. This is what it looks like in a web browser...

http://www.neu.edu.tr

If you click on it, it will send you to the main page of NEU. But what's that colored border around the image? That's what happens when you link an image. Personally, I think it's ugly. To get rid of it you must add a second, **BORDER** attribute to your image. Now the HTML for the image is...

And now it looks like...

http://www.neu.edu.tr

See? Much better. We set the **BORDER** attribute to "0" so it would not have a border. There are two attributes now in the **** tag. Does it matter what order the attributes go in? No. the **BORDER** attribute could be after the **SRC** attribute and it would still look the same. Now, when you click on the image above it still takes you to the main page of NEU, but there is no colored border around it.

Now you can add beautiful, intricate images to any of your web pages. In the next section I explain about placing images on your page by changing their alignment.

3.7 Aligning Images

Unlike text, aligning images doesn't require another element. All you must do is add another attribute. Ready? Let's go...

• Horizontal alignment

By default an image is aligned to the left. You can use **<CENTER> </CENTER>**) to center an image. To right-align an image you can use **<DIV ALIGN="right"> </DIV>** or you can do this...

3.8 Questions and Answers

Q. Do all elements have both an opening and closing tag?

A. No. Some just have one tag, but you'll learn more about them soon.

Q. Is the end tag just the start tag with the added slash?

A. Exactly.

Q. Will tags show up on the web page?

A. Nope. Anything between the <> will be hidden from the viewer.

Q. Should I capitalize tags like you did?

A. Although browsers don't care, it's a good idea. It helps keep tags separated from text and it's a good habit to start.

Q. Will text show up on the page even if it doesn't have a tag?

A. Yes, if you just type normal text into your HTML document, it'll show up like normal text on the web page.

Q. Can I use two tags at once?

A. Yeah, but you must remember to close both tags also. This is how it's done...

Chapter 4

Java

4.1 Coding By Java

Java is one of the most powerful languages used on the web. Java can be used not only for applets but for practical databases, search engines, & much more. Learning Java is difficult, but it can be done with some patience.

This section has been designed for users who may have little or no previous programming experience, but want to use Java on their web development projects.

At this point, this section covers only Java applets. An applet is a special type of Java program; it is primarily intended to be run within the context of another program, such as a browser. These are the types of Java programs that web developers would be interested in. You should, however, be aware that it is completely possible to design Java programs that do not require a browser to run. These types of Java programs are known as a Java applications. Although such programs are beyond the scope of this tutorial, applets are essentially a subset of Java applications. Thus the two share many similarities and learning one if you know the other is not difficult.

4.2 Why Java?

Java is an advanced programming language developed by the nice folks at Sun, and has many advantages over other languages.

The design of the language is such that it is extremely easy to learn; much easier than something like C++ (even though Java is just as powerful in many cases).

Java is the only language designed to be run (without any modification to the code) on virtually any platform. That is, the same programming instructions will work on a Macintosh computer, IBM PC, Sun workstation, etc. Unlike other programming languages, in which source code is directly compiled into machine specific language, Java compiles its programs into a format known as Java byte codes. Java byte codes can not be run without a program known as a Java "virtual machine." This virtual machine interprets the byte codes and then compiles the byte codes into machine specific languages. Thus when a Java program is transferred over the internet, what's really

25

being transferred are the Java byte codes, which are then ready to be interpreted by the virtual machine present on the host computer.

Java allows you to approach web projects with an unprecedented set of tools.

It's free! There is absolutely no need to go out and buy fancy tools to develop Java programs. All the tools you need can be downloaded over the internet; the specific tools are described in the next section of the tutorial.

4.3 How to start Coding by Java?

One of Java's greatest attributes is that all the tools you need to write, compile, test, and run Java programs are available for free. While there are certainly an abundance of commercial development tools on the market, the beginning programmer may feel intimidated by the price and complexity of these products. For our purposes, the free tools that Sun provides are more than sufficient. While at first they may seem slightly cumbersome, they are actually quite simple and straightforward. Use the following instructions to help you get set up for programming in Java.

(At this point in time, installation instructions for Java compilation tools are provided only for Windows based platforms)

The first thing you want to do is download the latest Java Development Kit (JDK). The JDK is a collection of tools that enable you to write Java applications and applets. It provides *all* the tools necessary for you to write your own Java programs from scratch. The JDK is a free product that is developed and maintained by Javasoft, a subsidiary of Sun.

The latest version of the JDK can always be found at http://java.sun.com/products/jdk/. Follow the appropriate links (look for the latest release of the JDK, but do *not* download JDK 1.2. Generally you will want to download a JDK of version 1.1.x). After downloading the file, open it from Windows and with any luck it should install without a problem.

At this point, the JDK should be functional. However, you may notice that it did not add a program group to the Start menu, nor did it put any sort of icon anywhere. This is because, for better or worse, the JDK is operated almost exclusively from the DOS prompt. Don't go nuts on me just yet, though. It's really not as bad as it sounds.

Before we proceed you may want to do one last thing. The JDK does not come with a source code editor. That is, unlike other fancy compilers, the JDK makes you responsible for somehow writing the source code into a text file. Thus unless you have a

26

particular affinity for Notepad, you may want to pick up a nicer, Java enabled editor. [Link coming]

If you've completed all these steps successfully, then your system should be ready to work with Java. Proceed to the next section to start your first real Java programming lesson!

4.4 Object Oriented Programming

You've probably heard of object oriented programming. This term is used to describe a very specific approach to solving problems and grouping data -- two very essential tasks in any programming endeavor.

Java is a completely object oriented programming (OOP) language. This probably doesn't mean much to you. Let's fix that.

Programming can be divided into two things: data, and the instructions performed on that data (also known as procedures or methods). In traditional programming languages (C, Pascal, BASIC), there is a sharp separation between these two entities. Object Oriented languages, however, are designed such that you must group the data and their methods together very closely. While this may require a certain degree of extra thought at the 'design' stage of programming, the benefits are huge.

Before going any further, we need to define a few terms and concepts:

Class - a way to group data and methods together into one coherent package. When you create a class, you are defining the blueprint for an *object*.

Object - a unique instance of a class. If a class is a blueprint for a building, think of an object as the actual building itself. Note that you can create many buildings out of the same blueprint, just as you can create many objects from the same class.

Let's consider an example. Don't worry, we don't need to actually program just yet; this will be a conceptual example. Don't worry about how you would actually *do* all this; just understand the design of the program.

Suppose you have a list of employees, and you need to store various data about them (salary, job description, etc.). Suppose you also need to calculate their weekly pay and various other figures.

Using the objected oriented methodology, here is what you may do:

Define a class called Employee that allows for the storage of the needed data figures. Remember that in this situation you wouldn't want to state what these figures actually are just yet. We're making the *blueprint* for an employee's data record, and thus we must leave any information in the class generic to all employee's. We can set values for the data later on.

We would also create procedures in the class that would compute the employee's salary and any other needed figures.

Once we have defined the class, we are ready to actually create objects. Each employee in the database would have their own object. For example, suppose we were creating Joe's record. We would tell the computer to create an object *named* Joe of *type* Employee. The computer now knows to allocate memory and resources such that the object "Joe" can be stored.

After this object has created, we need to manipulate it. Again, remember that when we defined the class we only created a "template" to put the appropriate values for an employee's data. Now that we have an actual object (Joe) that *implements* the class, we can fill in those blank spots. Thus Joe's object becomes unique and describes, well, Joe.

If we need to add another employee to the database, we can simply create another object of class Employee. We don't need to rewrite the code telling the computer to set aside space to store this employee's salary and other data; all this work is already accomplished when we defined class Employee. Instead, we just create another Employee object and manipulate it to our needs. We also don't have to rewrite the methods that are performed on the employees data; all this information is contained within class Employee.

4.5 Data

Storing data is an essential task in any program. While there are certain exceptions, almost any program ever made needs some sort of data to be useful. In programming, the way in which you store data is through *variables*.

Variable: a symbol given to a piece of memory that is used to store information.

There are several rules that go along with this definition. First of all, if you've declared that a specific variable (and hence it's corresponding portion in memory) will store numbers, then that variable can *only* store numbers as long as it exists. Thus you may be wondering what different types of variables you may use:

28

Data Type	Description
int	Numerical Integers. Values between negative and positive 2147483648
long	Numerical Integers. Has twice the range of values that an int does.
float	Decimal numbers. Values between negative and positive 3.4 x 10^38
double	Decimal numbers. Has twice the range of values that a float does.
char	One alphabetic character.
Boolean	A value of true or false.

The column on the left shows you the 'keyword' that Java uses to represent the different data types. Those are the keywords that you would put in your soure code to tell the compiler what type of variable you want. For example, to declare an *int* named 'salary', you would simply use the statement "int salary;". You probably noticed the semicolon sticking at the end of the statement. In Java, it is a required convention to end a complete programming instruction with a semicolon. This makes things easier on the compiler. Also, suppose you wanted to enter an initial value for that variable, such as "500". This is very simple, we simply modify the statement to read "int salary=500;". The equal sign **assigns** the value 500 to "salary", which is of type *int*. As usual, the statement is followed by a semicolon.

The difference between "int"s and "float"s are that integers are meant to store only positive or negative *whole* numbers. So if you wanted to store the number 3.14, an "int" would not be a good choice. A "float", however, would be an excellent choice since it is specifically designed to store decimal numbers.

You also may be wondering why all of the numeric data types (int, long, float, double) aren't consolidated into just the "double", since it could probably store any conceivable number you would use. The reason is that the larger the range that the data type has to cover, the more memory it will occupy in your system. While on today's machines this difference may seem neglible, it is considered good programming to only use the data type with the maximum range that you need.

4.6 Functions

In the last section you learned the basics behind data storage in Java. In this section, you'll get an introduction to *functions*. Functions are a way in which to logically organize the processes that your program performs on data.

Let's take a conceptual example first. Suppose you're writing a program that has to average 5 numbers that the user inputs. In order to simplify the actual coding, it is a good idea to list the tasks involved in going from the problem to the solution. For this example, you might write something like this:

- 1. Allocate (set aside) memory to store five numbers, an intermediate sum, and the final average.
- 2. Let the user input the five numbers and store them in memory.
- 2. Sum the five numbers.
- 3. Divide the sum by five to obtain the average.
- 6. Display the average to the user.

As you can see, we've split up a fairly simple task into six highly discrete, specific tasks. In programming terms, we can think of these tasks as separate functions. If you've had no programming experience whatsoever before, then you're probably wondering just what a function *really* is, and why you need them.

Think of a function as a black box. This black box has two holes, one for input and the other for output. The box doesn't necessarily have to be given input, nor does it necessarily have to output anything. But it provides the means to do so.

Now think of a program as a bunch of these black boxes strung together. In the example above, we could have six boxes, each one performing a corresponding task.

If we wanted, we could put *all* those tasks into one big box. For such a simple example as this, it would probably work fine. However, once you get into larger programs it is very important that you have as many black boxes as possible.

There is a very good reason for this. Suppose, in the program above, we wanted to output the average to a file instead of on screen. If the entire program was in one black box, we'd have to go mucking around in that big box and look for the code that handles that specific task. We run the risk of messing something else up in the process.

If, however, we only need to look at that *one* black box, the last one which handles output, we have absolutely no chance of messing up any of the other functions. All we

would have to do is reprogram that specific box and our program would keep on working.

When you have an appropriate amount of functions in a program, you have achieved *modularity*. A function (or black box) can be thought of as a module, and it's good to have a bunch of them. Keep that in mind.

One of the implicit consequences of this design is that you need someway to control the way in which boxes interact. After all, we can't have the function that displays the output operating before the function that inputs the numbers. In addition, we also can't allow the function that computes the sum to give it's output (incorrectly) to the function that gives the user output. All the boxes need to be controlled very precisely. This, of course, is built into the language and is extremely simple.

4.7 Operators:

Operators provide the means to actually calculate and compare things within your program. Simply put, an operator performs some sort of *operation* on one or more *operands*. Here are a few examples:

Operator	Description and Use
+	Adds two operands together
-	Subtract one operand from another
*	Multiplies two operands together
/	Divides one operand into another

These, of course, are the basic mathematical operators that Java provides. Consider the following code fragment:

int num1=5;

int num2=13;

int sum;

float average;

```
sum = num1 + num2;
```

average = sum/2;

Here we create three int variables. The first two, num1 and num2, are assigned values of 5 and 13 using the = *assignment operator*. This is a special type of operator; it literally

'assigns' the operand preceding it the value of the operand following it. In other words, in the case of "int num1=5;", the = operator assigns the value 5 to the newly created int variable num1.

There is another type of operator available to us called the *relational* operator. A relational operator compares its operands and returns a result based on that. This 'result' is always a *boolean* value. That is, the relation is either true or false. This makes sense since when you are comparing two things in terms of "is this larger than that?", the result can only be true ("yes, this *is* larger than that") or false ("no, this is *not* larger than that"). Thus consider these relational operators:

Sample Use	Description
a > b	Returns true when 'a' is greater in value than 'b'
a < b	Returns true when 'a' is less in value than 'b'
a >= b	Returns true when 'a' is greater than or equal in value to 'b'
a <= b	Returns true when 'a' is less than or equal in value to 'b'
a = b	Returns true when 'a' is equal in value to 'b'
a != b	Returns true when 'a' is not equal in value to 'b'

Relational operators are usually used with *conditional* statements. Conditional statements are the subject of another section. However, the basic idea them is, for example: "*if a is greater than b, then do this*".

4.8 Expressions:

Expression: any legal combination of symbols that represents a value.

Every expression must contain at least one operand and may use one or more operators. Operands represent values that operators use. Operators are keywords that represent specific actions that Java performs. For a more detailed discussion of operators, The "value" that the symbols represent is a value that is of any legal data type. For example, an expression can evaluate to an int, float, char, or any other legal data type (this too was discussed in a previous section). An expression can even evaluate to an object of a class you create. A class is really just a custom data type, and hence it is subject to the same abilities and restrictions as the standard data types.

At this point, this is all you need to know about expressions. You will automatically pick up the basics in terms of code as we progress; for now it is enough to simply

32
understand that when you have a bunch of Java statements together performing something, it always evaluates to *one* object or data type, which is then used by or assigned to something else.

4.9 Interface

An interface is an interesting Java feature that allows you to do a tremendous amount of things. In essence, an interface is a special class that defines a way to behave. An interface is fairly useless by itself. It only comes into play when another class *implements* the interface and its behavior.

An interface only defines methods and declares constants. The methods themselves do not have any code in them; the point of an interface is simply to enforce some sort of structure on any class that implements the interface.

A class may implement many different interfaces. This is an extremely useful attribute of interfaces in that it allows for a class to behave in many different ways, thus becoming extremely versatile.

Classes may specify which interfaces they implement by using the **implements** keyword. Consider the following code:

class rightPanel extends Panel implements Observer

In this example, we have class "rightPanel" extending the base class Panel. However, it also implements the behavior of class "Observer". This means that class rightPanel will implement (provide code) for any functions defined in Observer. We could even have this class implement more than one interface:

class rightPanel extends Panel implements Observer, MouseListener

Here we have class "rightPanel" still extending Panel, but implementing both "Observer" and "MouseListener". rightPanel would thus implement any functions in both Observer and MouseListener.

You will get a much better idea of how we can benefit from using interfaces, and what exactly they are in the coming sections and examples. You should, however, now have a basic awareness and understanding of what interfaces attempt to provide.

There is another type of class called an *abstract* class that is very similar to an interface. In an abstract class, as in an interface, methods are only defined (not implemented). However, instead of other classes *implementing* this class, they become *subclasses* of the abstract class and then implement those methods. Again, the point of this is that it enforces a certain type of structure on subclasses, and can be extremely useful in

33

creating an accurate data model of something. For example, consider something like furniture. Now, you've never had an object of "furniture." Instead, you've had objects of things that were a *type* of furniture. They shared certain attributes. Thus by using an abstract class, you can create a class hierarchy for furniture with many subclasses that define attributes specified by the base class.

4.10 Declaring Classes

As you've probably gathered by now, Java is all about classes. Everything is based on them and designed around their structure. Thus it is important to have a good working knowledge of how to *define* classes; that is, how to code all their separate components. The first part of a class is the class *declaration*. In this statement, you alert the compiler that you're going to be creating a class. You also tell the compiler about some of the attributes of this class. These attributes should generally be specified in a specific order. The following list will break down those attributes, and are listed in the order they should be specified.

- **Public** this keyword specifies that the class you are creating may be used by any "package." A package is a collection of classes; they will be discussed in more detail later. You should generally make your class public. Note that if this attribute is not specified, the compiler will assume the class is non-public.
- Abstract specifies that the class will be 'abstract.' You cannot have an object of an abstract class, you can only subclass to make other classes that can have objects. A class is non-abstract by default.
- Final specifies that the class cannot have subclasses. Used for specialized situations; only specify it if you know you need it. The compiler will assume the class to be non-final if this option is left out.
- Class Name Of Class Here after specifying whether the class is going to be public, abstract, and/or final, you use the keyword "class" and then the name you wish to use to refer to the class.
- Extends Name Of Base Class Here if this class is going to be a subclass, use the keyword "extends" and then specify the name of the base class.

34

• Implements Name Of Interface(s) Here - if this class is going to implement any interfaces, use the keyword "implements" and then the interface class names, separated by a comma if there is more than one.

Let's look at an example:

public final class NEU implements Runnable

Here we have specified a class called "NEU" (case is important) that can be read by other classes and packages (public), can not have any sub classes (final), and implements an interface known as "Runnable".

A curly-brace "{" follows the class declaration. Generally speaking, any block of code within a Java program is marked from beginning to end with curly braces, "{" and "}". For example:

public class NEU extends Applet

{

// code goes here

}

Here we have defined a class called NEU that is public and extends a base class Applet (something you will be doing for *all* of your applets). We then use the curly-brace to signify the beginning of the class code.

Then you see something new, "// code goes here". You probably guessed (correctly) that this doesn't actually mean anything to the computer. Any text followed by a "//" is ignored by the compiler. This used by programmers to provide themselves and other programmers with "comments" as to what they are doing.

In the following sections, we will further examine how to define class

Members - the methods and data that make up an actual class

Chapter 5

Graphics by PhotoShop

In this chapter you will find sections to get you started with PhotoShop, tricks to make your Please send comments, questions.

5.1 The Layers Palette

In this section, you will learn about the layers palette, what layers are used for, and how you can use layers.

5.1.1 Opacity

This is how transparent you want the layer to be. For example, say you have a layer that is a white rectangle on top of the background which is a green sandstone type material. By setting the opacity of the white rectangle layer down, you will allow more of the bumpy green background to show through, making the white less visible.

ala. Jaine wa		×
Layers	Channels Paths	
Normal		
T Preser	ve Transparency	1
3	Layer 1	a. (
3	Layer 1 copy	
3 /	Layer 1 copy 2	a dest
3	Backyround	1
5. Ur.		3



5.1.2 Preserve Transparency

If preserve transparency is on, then PhotoShop will only allow you to draw ON the layer. If it is turned off then you can draw anywhere on the image. For example, if you type some text and run the airbrush through it while preserve transparency is on, it will only draw on the text, whereas if you draw on it with the airbrush

while transparency is off, it will draw everywhere you do on the image(Fig. 5.1)



Figure 5.1



Chapter 5 – Graphics by PhotoShop 5.1.3 Why Use Layers?

Layers are very effective. Instead of having to take forever trying to select a certain part of the image, you can simply select that layer and apply filters and special effects to just that layer. Layers can also help you to create cool text effects and much much more. All photoshop users should get to know their layers palette well(Fig. 5.2).

5.2 The Toolbar

In this section, you will learn what tool each of the icons in the toolbox represents, and how to use that tool.

The Marquee Tool (M):

The marquee tool lets you select rectangular or elliptical areas by dragging over an area of the image. There are 3 different marquee styles that can be seen by when you hold the marquee button down. These styles include elliptical marquee, rectangular marquee, and a vertical and horizontal single-column marquee. To use, simply start at the top left corner of the area you want to select and drag down to the bottom right of the area.

P. The Lasso Tool (L)

The lasso tool lets you make a selection by dragging a freehand outline around an area. You can also constrain all or part of the outline to straight-edged segments. Click and drag around your selection area and simply let go to connect the endpoints and complete the selection. If you click the lasso button and hold it down, you can choose the straight edge lasso which "connects the dots" where you click, forming a polygonal selection.

The Magic Wand Tool (W)

The magic wand allows you to select an area of color. You can select the red pedal on a flower without having to trace around it. By adjusting the tolerance range, you can control just how sensative it is. If you choose a low tolerance, then the select ion area will have to be very close in color to where you clicked. Whereas, if you use a large tolerance value, the selection will not be as strict, and you will end up selecting a wider range of colors closely related to the red pedal.



Chapter 5 - Graphics by PhotoShop

The Airbrush Tool (A)

The airbrush tool works much like the paintbrush tool, but it creates much softer edges. To build up color, hold down the mouse button without dragging. By adjusting the pressure, you can darken or lighten the color. To change the brush, click the wind ow menu and select "show brushes", then click the brush you want.

The Paintbrush Tool (B)

The paintbrush lets you create soft strokes of color. If you turn wet edges on, it creates a watercolor effect where the paint builds up along the edges of the brush stroke. To change the brush, click the window menu and select "show brushes", then clic k the brush that you want to use.

The Eraser Tool (E)

The eraser tool changes pixels in the image as you drag through them. You can choose to change the color and transparency of the affected pixels, or to revert the affected area to its previously saved version.

The Pencil Tool (Y)

The pencil tool creates hard-edged freehand lines and is most useful for bitmapped images.

😤 The Stamp Tool (S)

The rubber stamp tool lets you paint a copy, or a modified copy, of an image or color into the same image or into another image. The Clone options of the rubber stamp tool make a copy of, or sample, an image and paint an exact duplicate of that image. Oth er rubber stamp options let you paint with a pattern or with an impressionistic copy of the image. You can also restore painted areas to their last-saved states.

1/2 The Smudge Tool (U)

The smudge tool simulates the actions of dragging a finger through wet paint. The tool picks up color from where the stroke begins and pushes it in the direction in which you

Chapter 5 – Graphics by PhotoShop drag.

(A) The Focus Tools (R)

The focus tools include the blur tool, which lets you blur hard edges or areas in an image to reduce detail, and the sharpen tool, which lets you sharpen soft edges to increase clarity or focus.

• The Toning Tools (O)

The toning tools include the dodge and burn tools, which let you lighten or darken, respectively, specific areas of an image, and the sponge tool, which lets you change the color saturation of an area.

4. The Pen Tool (P)

The pen tool is a tool that helps you create curved lines. To create a simple curve, always drag in the direction of the bump of the curve first, and then drag in the opposite direction. Dragging in the same direction creates an S curve. Hold the pen button down to reveal a few more buttons related to the pen tool.

T. The Type Tool (T)

This tool is probably the most widely used photoshop tool available. You can write plain text, by using the regular text tool, or do a text selection outline by holding the text tool down and selecting the text outline tool. Anti-aliasing helps to make your text blend smoothly with the background and get rid of those jagged edges.

N The Line Tool (N)

The line tool draws straight lines on an image. Line tool options let you specify the width of lines, set anti-aliasing, and create lines with arrowheads. To draw a line, select the line tool and drag in the image. To constrain the line angle to a multip le of 45°, hold down Shift as you draw.

The Gradient Tool (G)

The gradient tool lets you create a gradual transition between two or more colors. You

Chapter 5 - Graphics by PhotoShop

can choose from the existing gradient fills in the Gradient Tool Options palette, or you can create and edit your own gradient fills. If you don't select a specific part of the image to fill, the gradient tool applies the fill to the entire active layer.

The Paint Bucket Tool (K)

The paint bucket tool fills adjacent pixels that are similar in color value to the pixels you click. The paint bucket options let you specify a mode and opacity, indicate the color range, or tolerance, of pixels to be filled, and choose whether to fill wi th the foreground color or a pattern. You can also choose to create smooth edges for the filled selection.

The Eyedropper Tool (I)

The eyedropper tool lets you click a pixel (or group of pixels) on an image and set that color to be your foreground tool.

The Hand Tool (H)

You can view different areas of an image using the image window scroll bars or the hand tool.

Q The Zoom Tool (Z)

The zoom tool and the Zoom commands let you magnify and reduce the view of an image. You can also zoom to an exact percentage of the original view by entering values in the zoom percentage box at the lower left of the window. The image window's title bar displays the magnification or reduction percentage at all times. You can magnify up to 1600%.

5.3 Chrome Text



Make a new image with a black background. Create a new channel (channel #4) with your white image or text. Be sure to use a thicker font so the chrom e will stand out.



Duplicate the channel by dragging channel #4 onto the new layer icon, creating channel 5. Run a **Gaussian Blur** on the channel with a value of ab out 6.



Create a new channel (channel #6). Load the channel #4 selection (control+click channel #4), select channel five and **copy** the selection (contro l+c). Now **paste** it into channel 6 (control+v).



Your channels palette should look like this if you did the first three steps correctly. Channel #4 is the original text, channel #5 is the blurr ed text, and channel #6 is the trimmed text.



Create a new layer, and fill it with white. Now run the lighting effects filter on the image. make sure that 'Texture Channel:' is set to "#6". Or, You can see a screen shot of the light settings.

Chapter 5 - Graphics by PhotoShop



Choose layer 1, then **Load Selection** from channel #4, by control+clicking channel #4. Now **Invert the Selection** (control+shift+i), and **Clear** (backspace).



Now, **Invert** (control+i) the image, and give it a <u>drop shadow</u>. If you did it correctly, your image should loo k something like this, if not.. keep trying.

5.4.1 Glowing Text



Create a new layer with the image or text you want to glow.





Select the layer's transparency by right clicking the layer's thumbnail in the layers palette.

Chapter 5 – Graphics by PhotoShop

Expand the selection (select|modify|expand) by 2 (or more). This will decide how thick the glow is.



Create a new layer behind the main text layer, by choosing the background layer and clicking new layer. Choose that layer and fill the current selecti on with your glow color (white in the example). **Deselect** (control+d), and **Gaussian Blur** the layer with a value of 3 or so (whatever looks good to you.) Now adjust the hue/saturation if you desire.

5.4.2 How to Break an Image



Open the image you want to break up into seperate images. Make sure you have rulers turned on (control+r).

Note: if you are breaking y our image up horizontally, drag guides from the left ruler instead of the top ruler.



If you want to break your image up vertically like the example, click the top ruler and drag down where you want to make the first cut. This will crea te a blue guide. Do this for all of the divisions.



Select the rectangular marquee tool, and select from the top of the image to the blue guide. **Copy** the selection (control+c).

Create a new imag e (control+n). Paste it into the new image (control+v).

overview.

Repeat this process for the rest of the divisions.

5.5 Fading Images:



Select an image that you want to fade.

<u>W</u> idth:	4.167	inches
<u>H</u> eight:	2.167	inches
Anchor:		
	and an and a second sec	

Select white as your background color, and make the image taller with **Canvas Size** (Image|Canvas Size). Increase the height by about an inch whi le leaving the width the same. Make sure you click the little box in the top

middle



You should have about an inch of blank space at the bottom of your image that will be trimmed off later.

Chapter 5 - Graphics by PhotoShop



Use the lasso tool to draw around the part you want to fade at the top, but be sure you bring it all the way down to the bottom white part like i did in the picture.



Feather your selection by about 40, and Invert the Selection (control+shift+i).



Clear (backspace) the selection, and **Deselect** (control+d). You should still have the white space at the bottom of the image.



Using the crop tool, drag it from the top left corner to the bottom right corner of the IMAGE, not the white space. Once you finish cropping the image, the white space should be gone.

5.6 Animated Motion:



Select an object you would like to give motion to. The example uses a picture of the new VW Beatle.

Take into consideration how you want to make the object move. For instance, the example will be a horizontal looping motion. Although you might want to make your company's logo "fly-in" diagonally from the top left, or something to that effect.

		Ä
Layers	Anamels Paths Actions	
INormal	2	1
L Preser	ve transparency	Carlos -
	1 сору 8	
i.e.	1 copy 7	2019
	🚗 1 сору б	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	1 copy 5	115
	1 copy 4	
	🚗 1 сору 3	
	1 copy 2	
3 1	l copy	
3	bg	THE AL
- Hanna		N.I.S.

Next you'll have to duplicate the main image for the number of frames in the animation.

It is good to setup some method to the number of images. The example has 8 extra duplications, including the base image. This is because there will be 4 frames of the "drive-away" sequence, and 4 frames of the "drive-in" sequence.

pixels right
Thirty at Bade Avian Printing of
pixels down
<u> </u>

The third step involves placing each individual image's position. This can be done using the **Move Tool**, or a more exact method: the **Offset** filter.

The example shows the **Offset** filter, accessed through the filter menu: Filter / Other / Offset. It's good to check the dimensions of the image, to figure out which values (Horizontal or Vertical) to

change in each frame. The example image's width is **260**. The **52** is gotten through some simple mathematics. ((image width / framecount(+1)) * 2)

Lauers	hannels Paths Au	
Normal	- Opacity.	190%
Preserve	Transparency	
	📻 1 сору 8	1
27 E	1 copy 7	
	🖌 1 сору Б	
	1 copy 5	-
	1 copy 4	
5	1 copy 3	
	1 copy 2	-18 X
	1 сору	- p
3 / [
3	bg	
Construction of the second		J 🖬 🗸

After setting each frame's position, this is what the example's layer window looked like.



The last step is to get that added effect of speed and motion, using the **Motion Blur** Tool (Filter / Blur / Motion Blur).

Take into consideration the motion's path when setting the **Angle** property. The example has a

Chapter 5 – Graphics by PhotoShop

horizontal path, so therefor the Motion Blur Angle property is set to ZERO. It is good to have, again, a method to your bluring. The "fastest" frames should have the most blur, and each of the "frames" should have less of a blur.



Finally after applying the **Motion Blur**, throw each frame in an animator, and VOILA!

5.7 Rounded Corners

🗟 Unti	itled-1	@ 100%	(Layer 1, R	X
10	<u></u>	1	2	3
-	-			
-	1			
1				

Make a new layer and create a rectangle on it using the marquee tool and paint bucket.



so you can access it later.

Draw a circle using the elliptical marquee tool that is about half the height of the rectangle. Make sure you hold down shift while doing this so it c reates an even circle. Now you should **Save the Selection**

🖥 Untitled-1 @ 100%	(Layer 1, B 💶 🗙
	Untitled-1 @ 100% (Layer 1

Now select all of layer 1 except for the little corner piece in the top left.

Chapter 5 - Graphics by PhotoShop

Chapter 5 Chapmes of	y i notosnop
Te Untitled-1 @ 100%	(Layer 1, R 🖃 🔳 🗙
	Untitled-1 @ 100%
1	¹
Untitled-1 @ 1002	6 (Layer 1, R 📃 🗆 🗙
	2 3
δ	
and a state which which we shall be also and a state of the state of the	- 0. (2000) - 10000 - 10000 - 10000 - 10000 - 10000 - 10000 - 10000 - 10000 - 10000 - 10000

Now Invert the Selection

(control+shift+i), and then **Clear** the new selection. You have completed the first corner.

Load the Selection and repeat those steps for the rest of the corners.

Chapter 6 - Online Applications

Chapter 6



Online Applications

The concept of online computing has been talked about for years. There are many variations, but the basic idea is that a software application runs on a server, instead of on a local desktop machine. The client machine connects to the server over the Internet, but serves only as a user interface. This is technically known as a server-based computing model, and it offers several advantages, as we shall see. This form of computing is also called an online computing model, an online application, or a hosted application.

the moment, Application Service Providers (ASPs) are capturing the headlines, but they're only part of the story. Several forms of online applications have already been quietly going about their business for quite a while. The outlook for the future is nothing short of explosive. Together with other trends, including the growth of wireless and of broadband Internet access, the online application model points to a very different world ahead for Web developers.

6.1 Computing Models

When we speak of "computing models", we're basically talking about different ways of arranging computers into a network. Different models have come in and out of favor, usually driven by the kind of technology available at the time. For example, when small, powerful desktop computers became cheap and easy to use, mainframes started looking clunky and inflexible, and the client/server model was born. Now that Internet access is becoming more universal, the server-based, or online, model is becoming more attractive.

As you can imagine, each computing model actually has its own strengths and weaknesses. These are the main models in use today:

• Mainframe and Terminal - User machines (called "dumb terminals") are just I/O ports for a mainframe. They are permanently tied to the host network.

6.5 ASPs or ISP:

The end-user software products that we've mentioned so far provide not only the software itself, but the hosting service that runs it over the Internet. For larger organizations (and perhaps soon even for individuals), a third-party Application Service Provider (ASP) is another option.

ASP may offer software from traditional vendors, custom applications, or a mix of both. There are ASPs of every possible description, as well as "ASP aggregators", a sort of meta-ASP that packages the services of other ASPs and offers the much-coveted "comprehensive solution" to an organization. Someday, companies like this may offer a complete "turnkey" computing network, including everything from enterprise software to Office apps to data warehousing to phone service. For now, however, most of the general-purpose ASPs appear to be struggling, even as software vendors are having success with their hosted versions.

The reason is simply that the online model is far from mature. Many organizations and individuals are using some hosted applications, but few have moved their entire computing scene online. If you're going to use just a few online apps, then it makes sense to buy an "end-to-end" solution directly from the software vendor. If you're transferring your entire network online, however, then you'll want to deal with a single ASP.

Obviously, existing ISPs are gearing up to trade the "I" for an "A". Internet hosting has become something of a commodity business, with correspondingly low profit margins. Many ISPs are hungry for more profitable lines of business, and application hosting could be just that. An Internet hosting service is a "set it and forget it" service for most end users. If an ISP is doing a good job, the customer should seldom need to contact them, or even think about them. An ASP, however, is a different animal. Hosting critical applications for a customer means a lot of interaction with a customer, which logically should mean plenty of opportunities for lucrative consulting fees.

As with any hot trend, the hosted applications frenzy has spawned scads of start-up (or is that upstart?) companies hoping to cash in. "We'll start an ASP, and offer a full-service solution!" cry the young Turks as they soak up the VC cash. But the old guard of existing ISPs and big software vendors has not been snoozing - this isn't 1994 - and it's tough to see how there's going to be room for a bunch of new "pure ASP" companies

57

in this crowded and competitive market. Most or all of the organizations (and individuals) that are potential ASP customers already have a relationship with an ISP, and the ISPs are already starting to pitch the hosted applications hot and heavy. And, as we discussed above, many big software vendors are setting up as ASPs themselves.

As the online model matures, "Webtop" apps will look and feel just like normal software applications. For an end user, there would seem to be no discernable difference between a server-based and a client/server model. But what about for developers of Web content? This trend spells major changes for the Internet and applications development world, as we'll see in the <u>next section</u>.

6.5 Web Developer:

The online computing model is one of several trends that spell big changes for Web developers. The very concept of a Web site is changing rapidly. To some, it may seem that the question of a computing model is really of interest only to network administrators. At the moment, a lot of online apps run through a Web browser, which makes them clunky and very unlike traditional desktop apps. But this won't last. As the online model matures, "Webtop" apps will look and feel just like normal software applications. For an end user, there would seem to be no discernable difference between a server-based and a client/server model. But what about for developers of Web content?

The online application stampede is one of several related trends (the others include wireless and broadband) that are transforming the Internet into something far more than a mere purveyor of e-mail and magazine-style Web "pages". The line between Web developers and software developers is blurring quickly, and will soon disappear. Web weenies will find themselves getting more into the advanced languages like Java and C++, and using less of the Web-oriented scripting languages like JavaScript and Perl. Application developers will find themselves having to bone up on Internet blarp like TCP/IP and HTTP.

And I'll say it again: Everyone's going to have to <u>wise up to XML</u>. The Internet is quickly becoming the medium of choice for all electronic communications, encompassing video, applications, telephony and other stuff that we can't even imagine yet: Far more than the text and pictures that HTML handled so...adequately. We need a language that allows for any kind of content, and that is truly extensible, allowing future

58

enhancements to be made easily. EXtensible Markup Language (XML), a subset of SGML, is the current tool that provides these wonderful things. It's the basis of the popular Wireless Markup Language (WML), which is used to create content for cell phones and other wireless gadgets, as well as XHTML, a language used to update existing HTML content to an XML-compatible form.

The decline and fall of the operating system will also have a major impact on Web developers. In today's desktop world, a standardized OS is a huge asset (despite what the US Justice Department thinks). Forgive me, Mac-using friends, but if you use any other OS than you-know-what, you're adding a whole layer of compatibility problems to your workload. For better or for worse, this fact has had a huge impact on the whole development scene.

In the coming online world, that won't be so. In fact, which OS you choose to use will make not a whit of difference. Your client gadget will communicate with servers via TCP/IP, so you may use Mac, Linux, OS/2 or whatever you like. Maybe you don't even need an OS at all - just a Java Virtual Machine. This means two things. First, developers won't need to develop for different platforms. Thanks to the magic of XML and the online model, both content and applications will finally be write-once, read-anywhere, eliminating a lot of wasted work. Second, since no one OS will dominate the scene, no one company will be able to dictate how content and applications will be developed. A far greater range of development tools will be available, and developers will be able to do a lot of things that previously weren't possible because of OS-related limitations.

What about the tools we use to develop Internet content? These will migrate online too, but the impact of the shift may be subtle. Such things as editing video and audio would be nifty to do online, as the processor-gobbling rendering process would be done on powerful servers. But developing applications and HTML-type content will probably still feel pretty much the same. Endless online libraries of everything from clip-media to code modules will make the development world a vast playground for the creative mind. The online computing model will lead to an incredible array of new software tools, which will allow us to create even cooler software tools, and so on and on to computing Nirvana.

Chapter 7

NEU Students Online Active website:

7.1 Overview:

After we have read all previous chapters we should understand all the basic concepts of creating and designing a website. So in this chapter, I show my active application which is NEU STUDENTS ONLINE Website

And the main interface of this website as shown in (Figure 7.1) is consisting of 6 links and they are in order:

- Welcome
- Email service
- Chat Service
- Books Online
- Link & Search
- About



Figure 7.0

The main HTML file behind this interface is index.html and this index file is also compound of 17 other html files they are:

Bottom.html Bottomleft.html Bottomleft1.html Menu.html Right1.html Right2.html Right3.html Submenu.html Top.html Topleft.html Topmain.html Main1.html Main2.html Main3.html Main4.html Main5.html Main6.html

In this chapter also I describe the main html files which they are (main1.html, main2.html, main3.html, main4.html, main5.html, main6.html) (Fig.7.2)





7.2 Index.html File:

7.2.1 HTML Code:

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1256">
<meta name="Author" content="NEU Students Online.">
<meta name="Description" content>
<meta name="Keywords" content>
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title>NEU Students Online</title>
</head>
<frameset rows="45,*" framespacing="0" border="0"
frameborder="No">
  <frameset cols="345,*">
    <frame name="TOPLEFT" src="topleft.html" scrolling="No"
    marginwidth="0" marginheight="0" noresize
    frameborder="No">
    <frame name="TOP" src="top.html" scrolling="No"
    marginwidth="0" marginheight="0" noresize
    frameborder="No">
  </frameset>
  <frameset rows="*,30">
    <frameset cols="135,*">
       <frameset rows="252, *">
         <frame name="MENU" src="menu.html" scrolling="No"
         marginwidth="0" marginheight="0" noresize
         frameborder="No">
         <frame name="SUBMENU" src="submenu.html"
         scrolling="No" marginwidth="0" marginheight="0"
         noresize>
       </frameset>
```

62

```
<frameset cols="*,27">
```

<frameset rows="32,*">

<frame name="TOPMAIN" src="topmain.html" scrolling="No" marginwidth="0" marginheight="0" noresize frameborder="No"> <frame name="MAIN" src="main1.html" scrolling="Auto" marginwidth="0" marginheight="0" noresize frameborder="No">

</frameset>

<frameset rows="31,*">

<frame name="RIGHT1" src="right1.html" scrolling="No" marginwidth="0" marginheight="0" noresize frameborder="No"> <frame name="RIGHT2" src="right2.html" scrolling="No" marginwidth="0" marginheight="0" noresize frameborder="No">

</frameset>

</frameset>

</frameset>

<frameset cols="135, *">

<frame name="BOTTOMLEFT" src="bottomleft.html" scrolling="No" marginwidth="0" marginheight="0"

noresize frameborder="No">

<frameset cols="*,27">

<frame name="BOTTOM" src="bottom.html" scrolling="No" marginwidth="0" marginheight="0" noresize frameborder="No">

<frame name="RIGHT3" src="right3.html"

scrolling="No" marginwidth="0" marginheight="0"

noresize frameborder="No">

</frameset>

</frameset>

</frameset>

```
<noframes>
  <body>
  <script language="JavaScript">
<!-- begin script
var msgBan6, msgBan6b, string6 = "", stringPart6, delayBan6, ban6;
function banniere6(delay) {
 delayBan6 = delay;
 if (string6.length = 0) {
  string6 = " ";
  msgBan6b = msgBan6;
  stringPart6 = "";
 }
 else if (string6.length = 1) {
  while (msgBan6b.substring(0, 1) == "") 
   stringPart6 = stringPart6 + string6;
   string6 = msgBan6b substring(0, 1);
   msgBan6b = msgBan6b.substring(1, msgBan6b.length);
  }
  stringPart6 = stringPart6 + string6;
  string6 = msgBan6b.substring(0, 1);
  msgBan6b = msgBan6b.substring(1, msgBan6b.length);
 }
 else string6 = string6.substring(10, string6.length);
 window.status = stringPart6 + string6;
 ban6 = window.setTimeout('banniere6(delayBan6)',delay);
}
// end script -->
</script>
  </body>
  </noframes>
</frameset>
</html>
```

7.3 main1.html File:

7.3.1 Welcome Interface:



7.3.2 Main1 File HTML Code:

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=iso-8859-1">

<meta name="Author" content="NEU Students Online">

<meta name="Description" content>

<meta name="Keywords" content>

<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">

<title>NEU Students Online</title>

</head>

dody bgcolor="#FFFFF" text="#000000" link="#000080"

vlink="#800080" alink="#FF0000"

onload="msgBan6='NEU Students Online'; banniere6(100);"

onunload="clearTimeout(ban6)">

<script language="JavaScript">

<!-- begin script

```
var msgBan6, msgBan6b, string6 = "", stringPart6, delayBan6, ban6;
function banniere6(delay) {
 delayBan6 = delay;
 if (string6.length == 0) {
  string6 = " ";
  msgBan6b = msgBan6;
  stringPart6 = "";
 }
 else if (string6.length == 1) {
  while (msgBan6b.substring(0, 1) == "")
   stringPart6 = stringPart6 + string6;
   string6 = msgBan6b.substring(0, 1);
   msgBan6b = msgBan6b.substring(1, msgBan6b.length);
  }
  stringPart6 = stringPart6 + string6;
  string6 = msgBan6b.substring(0, 1);
  msgBan6b = msgBan6b.substring(1, msgBan6b.length);
 }
 else string6 = string6.substring(10, string6.length);
 window.status = stringPart6 + string6;
 ban6 = window.setTimeout('banniere6(delayBan6)',delay);
}
// end script -->
</script>
<table border="0" cellpadding="0" cellspacing="0" width="100%"
height="100%">
  \langle tr \rangle
     
  <img
```

src="neu.jpg" align="left" hspace="0" width="325" height="275">

<1>>

<applet code="jmText.class"</p> align="baseline" width="280" height="200"><param name="BGCOLOR" value="FFFFFF"><param name="FADEZONE" value="40"><param name="FONT" value="ARIAL"><param name="FONTSIZE" value="12"><param name="MODE" value="credits"><param name="REPEAT" value="yes"><param name="SPEED" value="80"><param name="TEXT1" value="Near East University"><param name="TEXT2" value="Computer Engineering Department"><param name="TEXT3" value=" "><param name="TEXT4" value="Graduation Project Year 2000"><param name="TEXT5" value="Submitted By"><param name="TEXT6" value="WAEL HIJAZI"><param name="TEXT7" value="SuperVisor"><param name="TEXT8" value="Miss. BESIME ERIN"><param name="TEXTCOLOR" value="000000"><param name="URL" value="http://NC.NEU.EDU.TR/~NEU ONLINE"><param name="VSPACE" value="3"></applet>

<form>

<font color="#808080" size="5"
face="Arial">search the internet using <select</pre>

name="liste1" size="1">

<option selected value>Search Engines </option>

<option value="http://www.yahoo.com">Yahoo </option>

<option value="http://www.alltheweb.com">all the web</option>

<option value="http://www.altavista.com">altavista </option>

<option value="http://www.netscape.com">netscape </option>

<option value="http://www.goto.com">goto </option>

</select> <input type="button" value="OK"

onclick="if (form.liste1.selectedIndex != 0) location =

form.liste1.options[form.liste1.selectedIndex].value; else alert('Please choose from menu.')">

</form>

</body>

</html>

7.4 main2.html File:

7.4.1 Email Service Interface:



Figure 7.3

7.4.2 Main2 File HTML Code:

<html>

```
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1256">
<meta name="Author" content="NEU Students Online">
<meta name="Author" content="NEU Students Online">
<meta name="Description" content>
<meta name="Keywords" content>
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
<title></title>
</head>
```

```
dody bgcolor="#FFFFFF" text="#000000" link="#000080"
vlink="#800080" alink="#FF0000">
```

```
height="100%">
```

```
<div align="center"><center><table
border="0" cellspacing="1" width="100%">
```

<11>

```
img src="emailko.jpg"
```

```
width="600" height="69">`
```

```
<a
href="http://neuonline.pinmail.com/"><img
src="login.jpg" border="0" width="113"
height="28"></a>
```

```
<a
    href="http://neuonline.pinmail.com/webprocess.fcg"><img
    src="signup.jpg" border="0" width="113"
    height="28"></a>
    
<1>
 <img src="at.jpg"</p>
 width="141" height="141"><img src="mail.jpg"
 width="333" height="170"><img src="at.jpg"
 width="141" height="141">
 <a
```

href="mailto:NEUSTAFF@neuOnline.pinmail.com">neustaff@neuonline.pinmail.com

```
</a>
```

```
</center></div>
```

</body>

</html>

7.4.3 Login Menu:



Figure 7.3.1

7.4.4. Signup Menu:

Step 1		
Please enter your personal det strictest confidence.	ails below. These details will be kept ir	1 the
Full Name:		
Street Address [1]:		
Street Address [2]:		
City/State:		
Country:	Select Location	
If "other" above:		
Your area's timezone:	GMT ??	
Telephone:		
E-mail address:		
Occupation:		Figure 7.3.2

Step 2	
Please select the neuonlin may be anything you choos other punctuation. The max	e e-mail address you would like to use. This e, but must not contain spaces, symbols, or any imum length of the address is 12 characters.
neuonline address:	@neuOnline.pinmail.com



ster	3.5
leas ne fii	se give us the password you would like to use with your account. On rst 20 characters will be recognised.
	Password:
	Confirm Password:
	Please read the terms of the agreement and click to indicate your agreement to proceed:
	Agree A the the NetAgree
BY THI AGI COI AGI ANI IN' UI TH	COMPLETING THE REGISTRATION PROCESS AND CLICKING E "I AGREE" BUTTON, YOU ARE INDICATING YOUR REEMENT TO BE BOUND BY ALL OF THE TERMS AND NDITIONS OF THE AGREEMENT BELOW. READ THE REEMENT CAREFULLY AS IT SETS FORTH YOUR RIGHTS D OBLIGATIONS WITH RESPECT TO USE OF THE MOMENTUM TERNET SOFTWARE AND SERVICES. IF YOU ARE NOT LLING TO BE BOUND BY THE TERMS AND CONDITIONS OF IS AGREEMENT, YOU SHOULD CLICK THE "I DO NOT REE" BUTTON.
MO	MENTUM INTERNET END USER LICENSE AGREEMENT

Figure 3.3.4

7.4.5 Compose an Email



Figure 7.3.6

7.5 main3.html File:

7.5.1 Chat Interface:



Figure 7.4

7.5.2 Chat Login Interfaces:



Figure 7.4.1


Figure 7.4.2



Figure 7.4.3

7.5.3 Main3 File HTML Code:

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=windows-1256">

<meta name="Author" content="NEU Students Online">

<meta name="Description" content>

<meta name="Keywords" content>

```
<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">
```

<title></title>

</head>

dody bgcolor="#FFFFF" text="#000000" link="#000080"

```
vlink="#800080" alink="#FF0000"
```

```
onload="msgBan6='NEU STUDENTS ONLINE'; banniere6(130);"
```

```
onunload="clearTimeout(ban6)">
```

```
<script language="JavaScript">
```

```
<!-- begin script
```

```
var msgBan6, msgBan6b, string6 = "", stringPart6, delayBan6, ban6;
```

```
function banniere6(delay) {
```

```
delayBan6 = delay;
```

```
if (string6.length == 0) {
```

string6 = " ";

```
msgBan6b = msgBan6;
```

```
stringPart6 = "";
```

```
}
```

```
else if (string6.length = 1) {
```

while (msgBan6b.substring(0, 1) == "") {

```
stringPart6 = stringPart6 + string6;
```

```
string6 = msgBan6b.substring(0, 1);
```

```
msgBan6b = msgBan6b.substring(1, msgBan6b.length);
```

```
}
```

```
stringPart6 = stringPart6 + string6;
string6 = msgBan6b.substring(0, 1);
```

```
msgBan6b = msgBan6b.substring(1, msgBan6b.length);
}
else string6 = string6.substring(10, string6.length);
window status = stringPart6 + string6;
ban6 = window.setTimeout('banniere6(delayBan6)',delay);
}
// end script -->
</script><div align="center"><center>
<table border="0" cellpadding="0" cellspacing="0" width="100%"
height="100%">
 <img
   src="chat.jpg" width="600" height="69">
   <table border="0" width="100%"
   bgcolor="#FFFFFF">
      
      <a
      href="chat.htm"><img src="NEUCHAT.jpg" border="0"
      width="308" height="254"></a>
       
     <11>
        
       <a href="chat.htm"><font
      color="#004080" face="Arial"><strong>NEU
```

```
CHAT</strong></font></a>
```

7.6 main4.html File:

7.6.1 Books Online interface:



Figure 7.5

C++ DataBase Java Java Script Web Design Networks Unix & Linux CGI & PERL	Other Books
Books Online	
JavaScript Manual of Style	Download
Laura Lemay's Web Workshop: JavaScript	Download
Special Edition Using JavaScript	Download
Teach Yourself JavaScript in a Week	Download
Teach Yourself JavaScript 1.1 in a Week, Second Edition	Download

Figure 7.5.1

7.6.2 Main4 File HTML Code:

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=windows-1256">

<meta name="Author" content="Visicom Media Inc.">

<meta name="Description" content>

<meta name="Keywords" content>

<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">

<title></title>

```
<STYLE TYPE="text/css">
```

<!--

BODY {

margin-top:0; margin-left:0; margin-right:0;

}

.lien {

position: relative; top: -5; color: #FFFFFF; font-size: 8pt; font-family: verdana; fontweight: bold; letter-spacing: -1pt; text-decoration: none;

}

.divise {

position: relative; top: -3; color: #C0C0C0; background: #000000; font-size: 12pt;

}

-->

```
</STYLE>
```

<STYLE TYPE="text/css">

```
<!--
```

BODY {

```
margin-top:0; margin-left:0; margin-right:0;
```

}

.lien {

position: relative; top: -5; color: #FFFFFF; font-size: 8pt; font-family: verdana; fontweight: bold; letter-spacing: -1pt; text-decoration: none;

```
}
```

.divise {

position: relative; top: -3; color: #C0C0C0; background: #000000; font-size: 12pt;

```
}
```

```
-->
```

</STYLE>

</head>

```
<body bgcolor="#FFFFF" text="#000000" link="#000080"
vlink="#800080" alink="#FF0000"</pre>
```

onload="msgBan6='NEU Setudents Online'; banniere6(130);"

```
onunload="clearTimeout(ban6)">
```

```
bgcolor="#000000">
```

<11>

```
<NOBR> <SPAN CLASS="divise"> </SPAN> <a
```

href="cplus.htm" class="lien"

```
onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"
```

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">C++

```
<SPAN CLASS="divise"> |</SPAN> <a href="database.htm" class="lien"
onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"
```

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">DataBase

```
<SPAN CLASS="divise"> |</SPAN> <a href="java.htm" class="lien"
onmouseover="netie(-1,4)?this.style.color='#FFF0000':null;"
onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">Java</a>
```

 | Java

```
Script</a> <SPAN CLASS="divise">|</SPAN> <a href="webprg.htm" class="lien"
```

class- nen

onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">Web

Design | <a href="networks.htm" class="lien"

```
onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"
```

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">Networks

```
<SPAN CLASS="divise"> |</SPAN> <a href="unix_linux.htm" class="lien"
```

onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">Unix

& Linux | <a href="cgi.htm" class="lien"

onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">CGI

& PERL | <a href="other.htm" class="lien"

onmouseover="netie(-1,4)?this.style.color='#FF0000':null;"

onmouseout="netie(-1,4)?this.style.color='#FFFFFF':null;">Other

```
Books</a> <SPAN CLASS="divise">|</SPAN> </NOBR>
```

```
<script language="JavaScript">
```

<!-- begin script

function netie(net, ie) {

if ((navigator.appVersion.substring(0,3) >= net && navigator.appName == 'Netscape' && net != -1) || (navigator.appVersion.substring(0,3) >= ie && navigator.appName.substring(0,9) == 'Microsoft' && ie != -1))

return true;

else return false;

```
}
// end script -->
</script>
```

```
bgcolor="#000000">
```

```
<img src="best%20copy.jpg" width="617" height="56">
```

```
<script language="JavaScript">
```

<!-- begin script

var msgBan6, msgBan6b, string6 = "", stringPart6, delayBan6, ban6;

function banniere6(delay) {

delayBan6 = delay;

if (string6.length == 0) {

string6 = " ";

msgBan6b = msgBan6;

```
stringPart6 = "";
```

```
}
```

else if (string6.length = 1) {

while (msgBan6b.substring(0, 1) = "") {

stringPart6 = stringPart6 + string6;

string6 = msgBan6b.substring(0, 1);

msgBan6b = msgBan6b.substring(1, msgBan6b.length);

stringPart6 = stringPart6 + string6;

string6 = msgBan6b.substring(0, 1);

msgBan6b = msgBan6b.substring(1, msgBan6b.length);

}

}

else string6 = string6.substring(10, string6.length);

window.status = stringPart6 + string6;

ban6 = window.setTimeout('banniere6(delayBan6)',delay);

}

```
// end script -->
```

</script>

```
<img src="1.jpg" width="375"
height="62">
```

<img src="3.jpg" width="375"

<img src="4.jpg" width="374"

 $\langle tr \rangle$

<img src="2.jpg" width="375"

height="65">

height="62">

height="62">

</body>

</html>

7.7 main5.html File:

7.7.1 Link and Search interface:



Figure 7.6

• News Links:



Figure 7.6.1

• Universities Main Menu Links:



Figure 7.62

• Palestine Universities Links:

USA Canada UK Turkey Jordan Palestine Pakistan

World Universities

Colleges and Universities - Palestine

Universities:

The Islamic University of Gaza (IUG)

Al-Quds University

Bethlehem University

Figure7.6.1

• Computer Magazines Main Menu:



Figure 7.6.3

• Top Links Main Menu:



Figure 7.6.4

• Free Services Main Menu:



Figure 7.6.5

• Free Screen Savers:



Figure 7.6.5.1

Mail List:





7.7.2 HTML Code:

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=windows-1256">

<meta name="Author" content="Visicom Media Inc.">

<meta name="Description" content>

<meta name="Keywords" content>

<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">

<title></title>

</head>

```
<body bgcolor="#FFFFFF" text="#000000" link="#000080"
vlink="#800080" alink="#FF0000"
onload="msgBan6="NEU Students Online'; banniere6(130);"
onunload="clearTimeout(ban6)">
<script language="JavaScript">
```

```
<!-- begin script
var msgBan6, msgBan6b, string6 = "", stringPart6, delayBan6, ban6;
function banniere6(delay) {
 delayBan6 = delay;
 if (string6.length == 0) {
  string6 = "";
  msgBan6b = msgBan6;
  stringPart6 = "";
 }
 else if (string6.length == 1) {
  while (msgBan6b.substring(0, 1) = "") {
   stringPart6 = stringPart6 + string6;
   string6 = msgBan6b.substring(0, 1);
   msgBan6b = msgBan6b.substring(1, msgBan6b.length);
  }
  stringPart6 = stringPart6 + string6;
  string6 = msgBan6b.substring(0, 1);
  msgBan6b = msgBan6b.substring(1, msgBan6b.length);
 }
 else string6 = string6.substring(10, string6.length);
 window.status = stringPart6 + string6;
 ban6 = window.setTimeout('banniere6(delayBan6)',delay);
}
// end script -->
</script>
<img src="NEUservice.jpg"</p>
    width="617" height="69">
```

<div align="center"><center>cellspacing="1" width="100%" bgcolor="#15628A">

child driving a car <div align="center"><center>

<11>

width="100%" bordercolor="#008080">

 $\langle r \rangle$

 <table

```
border="2" cellpadding="3"
```

```
cellspacing="0" width="400">
```

<form

action="http://www.searchalot.com/cgi-

bin/search/remotesearch.cgi"

method="get"
name="internet">

Search:

<font size="2"</pre>

face="Arial"><select name="location" size="1"> <option value="1">The Internet </option> <option value="4">Ask Jeeves </option> <option value="5">Searchalot </option> <option value="6">Altavista </option> <option value="8">Remarq Newsgroups </option> <option value="9">Dogpile </option> <option value="10">Euroseek </option> <option value="11">Excite </option> <option value="7">All The Web </option> <option value="13">Google </option> <option value="14">Government </option> <option value="15">HotBot </option> <option value="16">Infoseek </option> <option value="19">Lycos </option> <option value="21">Mamma </option> <option value="23">Microsoft </option> <option value="24">Netscape </option> <option value="27">Webcrawler </option> <option value="28">Windows Shareware </option> <option value="29">Yahoo! </option> </select> <td valign="top" rowspan="2"> Keywords:
 <input type="text" size="20" name="search" class="text"> <td valign="top" rowspan="2">

<td valign="bottom"

nowrap><input

type="submit"

value="Search"

class="submit">

</form>

<font

color="#FFFFFF" size="3"><marquee

align="middle"

width="50%">Welcome To NEU Students Services Online

</marquee>

</center></div>

<font color="#FFFF80" size="3"

face="arial">Join the Mailing List!<font</pre>

color="#FFFF80">

</center></div>

```
<a href="worldnew.htm"><img
src="wnews.jpg" border="0" width="120" height="30"></a>
<a href="universi.htm"><img
src="university.jpg" border="0" width="120" height="30"></a>
<a href="magazine.htm"><img
src="ico%20copy.jpg" border="0" width="120" height="30"></a>
</d>
</d>
```

```
<a href="free.htm"><img
```

```
src="freee.jpg" border="0" width="120" height="30"></a>
```

```
<img
src="allneu.jpg" width="610" height="130">
```

7.8 main6.html File:

7.8.1 About Interface:



7.8.2 HTML Code:

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=windows-1256">

<meta name="Author" content="Visicom Media Inc.">

<meta name="Description" content>

<meta name="Keywords" content>

<meta name="GENERATOR" content="Microsoft FrontPage Express 2.0">

<title></title>

</head>

dody bgcolor="#FFFFFF" text="#000000" link="#000080"
vlink="#800080" alink="#FF0000">

<11>

```
<img src="Graduation.gif"
width="60" height="60"><img src="graduation.jpg"
width="290" height="60">
```

⊲tr>

11/

```
<img
src="WAEL.jpg" align="top" width="169"
height="66">
<img src="BESIME.jpg"
width="187" height="68">
<img src="NICOSIA.jpg"
width="138" height="38">
```

```
</center></div>
```

```
 
</body>
</html>
```

7.9 Html Files Flowchart:



HTML FFILES

1.10 Interfaces Flowchart:

INTERFACES FLOWCHART



Conclusion

Connecting people and communication among them interchanging ideas, thoughts and cultural gains is the main concern of recent and future technology.

Having apart or playing a role in this fast developing communication technology is not so difficult as it is thought of. If we only try to understand the philosophy of information technology and deal with the available tools. We can move forward from being end users to be creators in the new age of information and communicating technology.

Working in this project helped me so much in web designing field and specially in graphics work, Java and Visual Web Editors.

References

http://www.netpedia.com

http://www.netpedia.com/html/reference/ http://www.jsworld.com/ http://javascripts.earthweb.com/ http://shabaka.hypermart.net http://webdeveloper.com http://kitap.selcuk.edu.tr