Using Web Design Techniques and Tools in Developing Mathematical Software and Applications

A Project Report submitted as per requirements of unit MS-607 of the Course
M.A. / M.Sc. in Mathematics of Tezpur University

Session : 1999 – 2001

Submitted by
Himadri Barman
Department of Mathematical Sciences
Tezpur University

Guided by
Dr. Kishore Kumar Das
Senior Lecturer
Department of Mathematical Sciences
Tezpur University

Department of Mathematical Sciences
School of Science and Technology
Tezpur University
Tezpur – 784028
Assam, India

Project Title: Using Web Design Techniques and Tools in Developing Mathematical Software and Applications

Abstract: A revolution is taking place in the field of Information Technology. The Internet which began in the mid 1960s as a project of the United States Department of Defense's Advanced Research Project Agency, has today become a medium that connects every nook and corner of the world. One of the most visible aspect of the Internet is the World Wide Web (also called "W3", "WWW" or the "Web" for short). It emerged in the 1990s as a result of a new Internet Protocol called HTTP (HyperText Transfer Protocol). Simply speaking, the Web is a vast network of HTTP servers sending files across the Internet.

This minor project is a humble attempt to harness the power of the Web in the context of Mathematics. For this I concentrate on the publishing language of the Web, HTML (HyperText Markup Language) and JavaScript, an interpreted Web Programming Language. As a practical demonstration of what I have learnt, I design the Home Page of the Department of Mathematical Sciences. The idea behind my project is exemplified by a handy Web-based Calculator.

Keywords: Web Design, HTML, JavaScript, Home Page, Web Calculator.

Organization: Department of Mathematical Sciences
Tezpur University
Tezpur - 784028

Department of Mathematical Sciences
Tezpur University
Nappam
Tezpur 784028

CERTIFICATE

This is to certify that the project entitled “**Using Web Design Techniques and Tools in Developing Mathematical Software and Applications**“ was carried out by Himadri Barman under my supervision and guidance as per requirements of unit MS-607 of the M.A./M.Sc. course in Mathematics. He was sincere and hard working during the course of this project. I wish him all the success.

Date: 27.11.2000	Kishore Kumar Das
Place: Tezpur	Senior Lecturer

ACKNOWLEDGEMENT

During the course of my project "**Using Web Design Techniques and Tools in Developing Mathematical Software and Applications**", I received the guidance, help and encouragement from many. This acknowledgement is a small tribute to all of them.

At the very outset, I would like to sincerely thank the Head, Department of Mathematical Sciences, Dr. M. Borah, for giving me a free hand to choose my project guide as well as my area of interest to carry out the project work. I am equally thankful to my guide, Dr. Kishore Kumar Das, for giving me the opportunity to work in such a new area. Without his help and guidance, it would have been difficult for me to complete my project. I am also thankful to the other faculty members of my Department, especially Mr. Debajit Hazarika for giving me valued suggestions from time to time.

I am thankful to my classmates for their continued support and encouragement. Lastly I offer my sincere thanks to the students of the Department of Computer Science, Tezpur University for their help and support.

Himadri Barman
M.Sc. Third Semester
Department Of Mathematical Sciences
Tezpur University

Contents

CHAPTER 0
Introduction

I started this project with the objective of designing the home page of the Department of Mathematical Sciences, Tezpur University. But it was always in my mind that as a mathematics student, it would be better if I could use my knowledge of Web designing in some way related to mathematics. That is how my project took shape.

The title of my project sounds rather confusing. Why should one use Web designing tools to develop mathematical software and applications when a host of other proven programming tools are available for the same job. Let us see if there is any merit behind the whole exercise. We begin by developing a very simple program – to find the factorial of a number, first in C++ and then in HTML/JavaScript (a Web Authoring Tool).

The program (factorial.cpp) in C++ syntax is:

```
#include<iostream.h>
#include<conio.h>
void findFactorial(int x)
 {
   if(x<0)
    cout<<"Check Input";
   else if(x==0 || x==1)
    cout<<"The factorial of "<<x<<" is 1.";
   else
    {
     long int factorial=1;
     for(int i=1;i<=x;i++)
      factorial*=i;
     cout<<"The factorial of "<<x<<" is "<<factorial<<".";
    }
 }
void main()
 {
  clrscr();
  int x;
  cout<<"This program finds the factorial of a
  number"<<endl;
  cout<<"Enter the number:";
  cin>>x;
  findFactorial(x);
  getch();
 }
```

The program (factorial.html) in HTML/JavaScript syntax is:

```
<HTML>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!-- Hide
function findFactorial()
 {
  var inputNum=document.form1.input.value
  var factorial=1
  if(inputNum<0)
   alert('Check Input')
  else if(inputNum==0 || inputNum==1)
   document.form1.answer.value=factorial
  else
   {
    for(var i=1;i<=inputNum;i++)
     factorial*=i
    document.form1.answer.value=factorial
   }
 }
// End Hide -->
</SCRIPT>
<H3>FACTORIAL OF A NUMBER</H3>
<FORM NAME="form1">
Enter Number:
<INPUT TYPE="text" NAME="input" SIZE="3">
<INPUT            TYPE="button"            VALUE="CALCULATE"
onCLICK="findFactorial()">
<BR>
The Factorial is:
<INPUT TYPE="text" NAME="answer" SIZE="25">
</FORM>
</BODY>
</HTML>
```

Now, we look at some statistics. The file sizes of factorial.cpp, factorial.obj and factorial.exe are 528, 14358 and 31596 bytes respectively totaling 46488 bytes while the file size of the file factorial.html file is only 697 bytes. Even if we take only the factorial.exe file into consideration, there is a difference of 30899 bytes between the conventional C++ program and the technologically new HTML/JavaScript (Web) program or for that matter the C++ program is more than 45 times bigger than the Web program. Reason enough to try this new technology.

Let's explore further. To develop a C++ program, we will need a compiler and to develop a Web program, what we need is a simple text editor and to test it, a browser, both of which are part of a standard operating system like the dominant Windows 95/98. The essence is that no extra investment is needed.

A single Web program can be made to reach to a much wider user base at a minimum cost via the Internet or for that matter an intranet. Using a web program is thus much more cost-effective. And as far as aesthetics are concerned, a Web program can be given a cool look with the minimum of effort as compared to a program designed using the traditional tools.

Hardware Used:

Desktop: Pentium® based/75 MHz – 300 MHz/16-32 MB RAM
Scanner: Hewlett Packard ScanJet 4C

Software Used:

Browsers: Microsoft® Internet Explorer 3.0/4.0/5.0/5.5
Netscape® Communicator 4.0/4.6/4.7
Opera® 4.02 Unregistered version
Editor: Microsoft® Notepad for Windows95/98
Image Editing Tool: IrfanView 2.97

CHAPTER 1
Web Design

Web Design in simple terms is a methodology that is being used in the development of the Web, with the help of software called Web Authoring Tools (the software used to create Web pages). For simplicity, Web Design simply refers to the authoring of Web pages. Web pages are written in a computer language that gives general guidelines about the desired presentation; and as of now this language is called the HyperText Markup Language or HTML.

Interestingly, a person who creates a Web page does not need to understand the details of computer display hardware, nor does an author need to prepare a separate copy of the page for each type of computer display screen. Instead, a Web page is created in a general-purpose manner that can be used with any display hardware. Thus, an author needs to prepare a single copy of each Web page – instead of including all details in the specification about how to display the page on various types of computer hardware, the author uses a computer language that specifies the content of the page.

HTML even with its advantages falls short of providing the control many developers want when creating Web pages and applications. It led to the emergence of host of new technologies and the widespread use of both server side programming and client side programming. Server side programming can be done using two standard interfaces:

1. CGI (Common Gateway Interface).
2. NSAPI (Netscape Server Application Programming Interface) developed by Netscape and ISAPI (Internet Server API) developed by Microsoft.

CGI programming can be done using any compiled programming language like ForTran, C/C++ and the like. Specific languages for programming CGI scripts are available like Perl (Practical Extraction and Report Language) and Tcl (Tool Command Language). LiveWire and ASP (Active Server Pages) are the NSAPI and ISAPI tools respectively

Client side programming can be done using scripts such as JavaScript and VBScript.
Many other technologies have evolved over the past few years, which have helped in the evolving of Web design as a full-fledged discipline. The foremost among them is Java, a programming language developed at SUN Microsystems. Though not originally developed as a Web programming language, it is used to design Web applications called Applets. These applets are then incorporated in a Web page. The applets work in a pre-defined area of a Web page and contribute to the dynamic nature of the Web page.

Another exciting technology is the Virtual Reality Modeling Language (VRML). VRML helps give the Web a “there” feeling. It creates athree dimensional “there” where one can move around and explore. Just as the Windows desktop metaphor changed the way we use computers, the VRML 3-D metaphor is changing the way we use the Internet.

The incorporation of scripts and applets to make the Web pages dynamic have resulted in the emergence of what is called DHTML (Dynamic HTML). It has led to the introduction of a new standard called DOM (Document Object Model), which provides ways for scripts to manipulate HTML using a set of methods and data types defined independently of particular programming languages or computer platforms. It forms the basis for dynamic effects in Web pages, but can also be exploited in HTML editors and other tools by extensions for manipulating HTML content. A related DHTML technology is the CSS (Cascading Style Sheets). CSS provides a simple means to style HTML pages, allowing us to control visual and aural characteristics; for instance, fonts, margins, line-spacing, borders, colors, layers and more.

The next generation XML (Extended Markup Language) which is a cousin of HTML has already come into use. It allows us to define our own mark-up formats when HTML is not a good fit. Work on the next generation style sheet language written in XML called XSL (Extensible Stylesheet Language), which provides a means to transform XML documents into HTML is also on at a frantic pace.

Web designing, can thus be concluded as a culmination of various technologies with the ultimate aim of convergence in mind, the ultimate benefactor of which will be man himself.

CHAPTER 2
HTML: A Brief Overview

HTML stands for HyperText Markup Language. Hypertext is ordinary text that has been dressed up with extra features, such as formatting, multimedia, and links to other documents. Markup is the process of taking ordinary text and adding extra symbols. Each of the symbols used for markup in HTML is a command that tells a browser how to display the text. Markup can be very simple, or it can be very complicated. Either way, the underlying text being markedup is always present and viewable. Language is actually a key point to remember about HTML. HTML is a computer language, related to other programming languages. HTML has its own syntax, slang, and rules for proper communication. Markup languages are a special type of computer language because they are solely concerned with classifying the parts of a document according to their function . It's not really correct to speak of "programming HTML", because HTML isn't really a programming language. HTML is a markup language that has a different goal than creating a program.

HTML is neither a page-layout language nor a printing language. The only thing HTML does is classify parts of a document so that a browser can display it correctly. This allows documents to be displayed in many kind of platforms, not just ordinary personal-computer platforms like PCs and Macintoshes, but HTML is also designed to work on a wide variety of graphical workstations, "dumb" (that is, text only) terminals, network computers, hand-held devices, and much more. This means that HTML can be used to put a document on not just computer screens, but also printers, fax machines, TV sets, game consoles, Braille devices, digital watches, and text-to-speech machines.HTML and the Web were first conceived in 1989 by a researcher named Tim Berners-Lee who worked for CERN, the European Laboratory for Particle Physics in Geneva, Switzerland. CERN researchers developed the first World Wide Web programs in 1990 and began releasing programs and specifications to Internet users in 1991. The Web pioneers, led by Berners-Lee regrouped into the World Wide Web Consortium (W3C) in December,1994. The W3C is now responsible for the standards of HTTP, HTML, and other Web technologies.

HTML is one member of a family of markup languages called SGML, which stands for Standard Generalized Markup Language. SGML was developed by the International Organization for Standards in 1986 to define markup languages designed for various different purposes.

HTML code is written using special commands called tags. A tag is a coded command used to indicate how part of a Web page should be displayed. They are words starting with < and ending with > and are considered coded commands. Most, but not all, HTML tags have two parts: an opening tag < >, to indicate where a piece of text begins, and a sometimes optional closing tag </ >, to show where the piece of text ends. The tags coming in pairs are called surrounding tags, while those not coming in pairs are called separating tags. Both surrounding and separating tags make use of attributes and their attribute values, if any, are included in the tag. The integral part of each HTML document is the <HTML> tag. Some tags are used to specify the title of a document, others are used to identify headings, paragraphs and hyperlinks. Still others are used to insert forms, images, multimedia objects, and other features in Web documents. It is to be remembered that tags can be written in the upper as well as the lower case. Thus <HTML> and <html> are same. Below is a simple HTML file with the related concepts:

```
<HTML>
<!-- This is the comment tag -- >
<HEAD>
<TITLE>This text is the document’s title</TITLE>
</HEAD>
<BODY>
<P>This is a paragraph heading</P>
<HR SIZE=”10”>
</BODY>
</HTML>
```

In the above file, <P> is a surrounding tag, while <HR> is a separating tag with attribute SIZE and attribute value 10.

Although HTML is periodically standardized, the language continue to grow as the result of new tags, attributes, and attribute values that get introduced by the browser developers. These additions are not part of official HTML language, so they are referred to as extensions. Most extensions are eventually integrated into the official version of HTML. The latest official version is HTML 4.01.

As the role of the computer evolves, HTML is becoming more and more central to nearly everything we do with computers. HTML is the global standard for connecting all types of information together in a predictable and presentable way. HTML is the glue that holds together all the diverse types of information on our computers and ensures that it can be presented in a standard way that will look the same to anyone in the world. No wonder, HTML has been aptly called the Universal GUI (Graphic User Interface).

CHAPTER 3
JavaScript: A Brief Overview

JavaScript is a script based programming language that supports the development of both client and server components of Web-based applications. On the client-side, it can be used to write programs that are executed by a Web browser (this is what I am interested in as far as this project is concerned). On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser's display accordingly.

The script can perform actions, such as generating HTML code that affects the display of the browser window. It can perform actions that affect the operation of plug-ins, Java applets, or ActiveX components. The script can also define JavaScript language elements that are used by other scripts.

Some scripts may define functions for handling events that are generated by user actions. For example, we might write a script to define a function for handling the event "checking input data". The event handlers can then perform actions to validate the users input data.

One of the most important features of JavaScript is that it is an object-based language. This simplifies the design of JavaScript programs and enables them to be developed in a more intuitive, modular, and reusable manner. An object consists of two things:

- A collection of properties that contain data
- Methods that enable operations on the data contained in those properties

When we view something as an object, we look at it in terms of its properties and methods. Let us take the example of a Web page and consider it as an object (cyber-world object). Its title, bgColor, links, etc. would be its properties while open(), close(), write(), etc. would be its methods.

Often, one programming language will evolve from another. For example, Java evolved from C++, which evolved from C, which evolved from other languages. This is also the case for JavaScript.

Netscape originally developed a language called LiveScript to add a basic scripting capability to both Navigator and its Web-server line of products. When it added support for Java applets in its release of Navigator 2, Netscape replaced LiveScript with JavaScript. Microsoft introduced its version of JavaScript, referred to as JScript, in Internet Explorer 3. JScript is tightly coupled to Internet Explorer and allows almost all HTML elements to be scripted. JScript is compatible with JavaScript 1.2. Netscape and Microsoft submitted their scripting languages to the European Computer Manufacturers Association (ECMA) for standardization. ECMA released the Standard ECMA-262 in June,1997. This standard describes the ECMAScript language, which is a compilation of the best features of JavaScript and JScript. An updated version of this was released in June,1998. JScript 3.1 and JavaScript 1.3 are compliant with ECMAScript. Another JavaScript compatible browser is the Opera browser launched by Opera Software.

JavaScript statements can be included in HTML documents by enclosing the statements between an opening <SCRIPT> tag and a closing </SCRIPT> tag. Within the opening tag, the LANGUAGE attribute is set to “JavaScript” to identify the script as being JavaScript.

```
<SCRIPT LANGUAGE=”JavaScript”>
JavaScript Statements
</SCRIPT>
```

The script tag may be placed in either the head or body of an HTML document. In many cases, it is better to place the script tag in the head of a document to ensure that all JavaScript definitions have been made before the body of the document is displayed. Below is a simple JavaScript listing.

```
<HTML>
<HEAD>
<TITLE>A JavaScript Listing</TITLE>
</HEAD>
<SCRIPT LANGUAGE=”JavaScript”>
 document.write(“This is a JavaScript statement”)
</SCRIPT>
</BODY>
```

```
</HTML>
```

As can be seen, JavaScript is dependent on HTML. No JavaScript programs can be developed independently. Thus it is imperative that we learn HTML if we are to program in JavaScript. A browser interprets the JavaScript statements in the HTML file and displays the output in the browser window.

The script tag itself provides an alternative way to include JavaScript code in an HTML document, via the tag'sSRC attribute, which may be used to specify a file containing JavaScript statements. Here's how we do it:

```
<SCRIPT LANGAUGE="JavaScript"SRC="abc.js">
</SCRIPT>
```

In the above example, the file abc.js is file containing JavaScript statements. The file extension must be js.

JavaScript is thus a new and powerful programming language for Web designing purposes. It not only enables the development of truly interactive Web pages, it is also the essential glue that integrates Java applets, ActiveX controls, Browser plug-ins, server scripts, and other Web objects, permitting developers to create distributed applications for use over the Internet.

CHAPTER 4
JavaScript's Math Object

The Math object is a predefined JavaScript object which provides an extensive library of mathematical functions and constants. It is a built-in object that is defined by the ECMAScript standard and is supported by a host of web browsers. It is always available anytime a JavaScript browser is running. Learning to use the Math object involves familiarizing ourselves with these properties and methods.

Mathematical Constants:

For a scripting language, JavaScript provides a rich collection of mathematical constants tabulated below:

Constant	Description
E	Euler's constant, the base for natural logarithms.
LN2	The natural logarithm of 2.
LN10	The natural logarithm of 10.
LOG2E	The base 2 logarithm of E.
LOG10E	The base 10 logarithm of E.
PI	The ratio of the circumference of a circle to its diameter.
SQRT1_2	The square root of ½.
SQRT2	The square root of 2.

The mathematical constants are used as Math.constant where constant is one of the constants listed above. For example, pi times Euler's constant is expressed as Math.PI*Math.E.

Mathematical Functions:

The Math object provides 18 mathematical functions that run the gamut from rounding to random numbers generation. These functions are arranged in the following categories.

Ø Rounding
Ø Comparison
Ø Algebraic
Ø Logarithmic and exponential
Ø Trigonometric

Ø Random number generation

To use any of the mathematical functions, we invoke them as methods of the Math object. For example, the cosine of pi divided by 2 is written as Math.cos((Math.PI)/2).The categories of math functions are described in the following paragraphs.

Rounding Functions: The round(), floor(), and ceil() functions are used to approximate floating-point numbers with integers. The round() function returns the closest integer to a floating-point number. The floor() function returns the greatest integer that is less than or equal to a floating point number. Similarly, the ceil() function returns the least integer that is greater than or equal to a floating point number.

Comparison Functions: The min() and max() functions are used to compare two numbers. The min() function returns the lesser of the two numbers while max() function returns the greater.

Algebraic Functions: The abs() function calculates the absolute value of a number. The sqrt() function calculates the square root of a number.

Logarithmic and Exponential Functions: The natural logarithm function log() is provided. Base 2 or 10 logarithms can be calculated using the logarithmic constants (LN2, LN10, LOG2E, LOG10E) to convert between different bases. The pow() function calculates a number raised to a power. The exponential function, exp() calculates Euler’s constant raised to a power.

Trigonometric Functions: Seven trigonometric functions are provided, viz., cos(), sin(), tan(), acos(), asin(), atan(), atan2().

Random Number Generation: The random() function generates a pseudorandom number between 0 and 1.

APPENDIX 1
Source Code of the Web Calculator

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 //EN">
<HTML LANG="EN">
<HEAD>
<META NAME="Contents" CONTENT="Source Code of Web Calculator">
<TITLE>The Web Calculator : Beta Version 1.0</TITLE>
<SCRIPT LANGUAGE="JavaScript"><!--
r = new Array(2)
function setStartState(){
 state="start"
 r[0] = "0"
 r[1] = "0"
 operand=""
 ix=0
}

function addDigit(n){
 if(state=="gettingInteger" || state=="gettingFloat")
  r[ix]=appendDigit(r[ix],n)
 else{
  r[ix]=""+n
  state="gettingInteger"
 }
 display(r[ix])
}

function appendDigit(n1,n2){
 if(n1=="0") return ""+n2
 var s=""
 s+=n1
 s+=n2
 return s
}

function display(s){
 if(r[ix].length>17){
  alert('Outof Range')
  clearDisplay()
 }
 else document.calculator.total.value=s
}
```

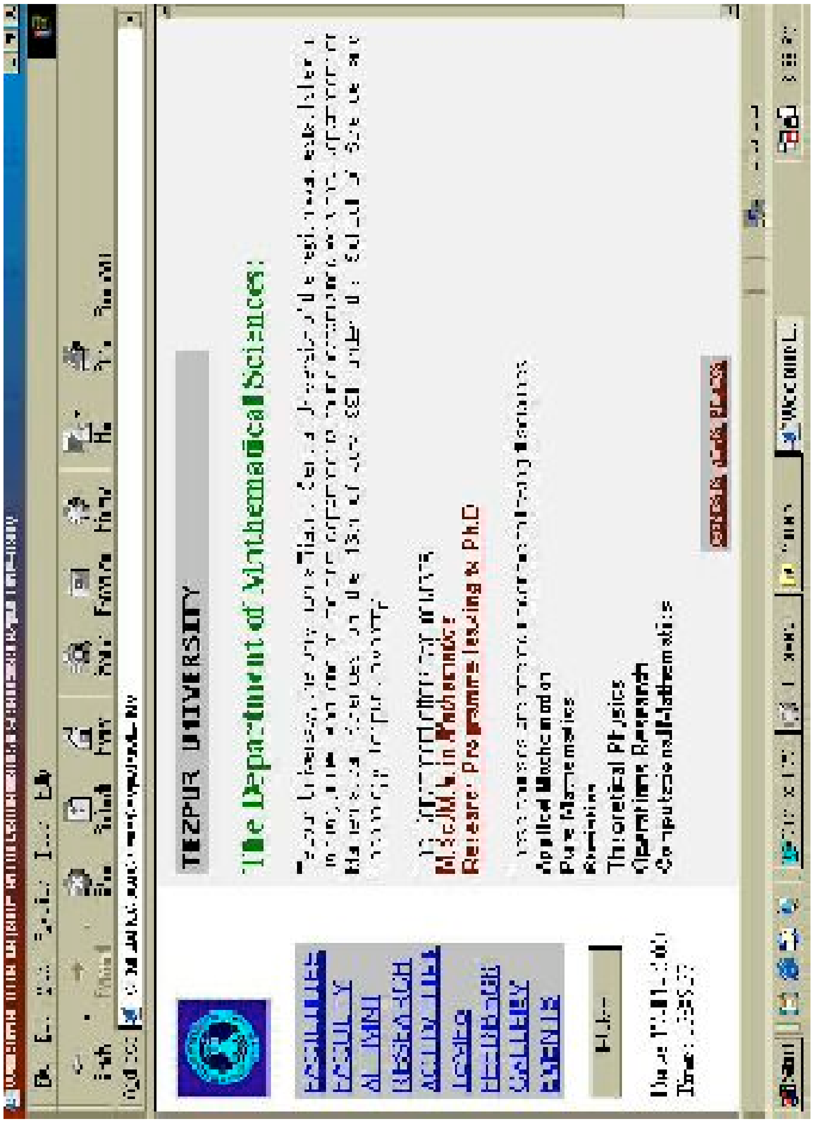
```
function displayResult(s){
  document.calculator.total.value=s
}

function addDecimalPoint(){
 if(state!="gettingFloat"){
  decimal=true
  r[ix]+="."
  if(state=="haveOperand" || state=="getOperand2") r[ix]="0."
  state="gettingFloat"
  display(r[ix])
 }
}

function clearDisplay(){
 setStartState()
 display(r[0])
}

function changeSign(){
 if(r[ix].charAt(0)=="-") r[ix]=r[ix].substring(1,r[ix].length)
 else if(parseFloat(r[ix])!=0) r[ix]="-"+r[ix]
 if(state=="changeSign")
  state="getOperand2"
 else
  state="changeSign"
 display(r[ix])
}

function setTo(n){
 r[ix]=""+n
 state="haveOperand"
 decimal=false
 displayResult(r[ix])
}

function calc(){
 if(state=="gettingInteger" || state=="gettingFloat" ||
  state=="haveOperand"){
  if(ix==1){
   r[0]=calculateOperation(operand,r[0],r[1])
   ix=0
  }
 }else if(state=="getOperand2"){
  r[0]=calculateOperation(operand,r[0],r[0])
  ix=0
 }
 state="haveOperand"
 decimal=false
 displayResult(r[ix])
}

function calculateOperation(op,x,y){
 var result=""
 if(op=="+"){
   result=""+(parseFloat(x)+parseFloat(y))
  }else if(op=="-"){
   result=""+(parseFloat(x)-parseFloat(y))
  }else if(op=="*"){
   result=""+(parseFloat(x)*parseFloat(y))
  }else if(op=="/"){
   if(parseFloat(y)==0){
    alert("Division by 0 not allowed.")
    result="ERROR"
   }else result=""+(parseFloat(x)/parseFloat(y))
  }
}

function performOp(op){
 if(state=="start"){
  ++ix
  operand=op
 }else if(state=="gettingInteger" || state=="gettingFloat" ||
  state=="haveOperand"){
  if(ix==0){
   ++ix
   operand=op
  }else{
   r[0]=calculateOperation(operand,r[0],r[1])
   displayResult(r[0])
   operator=op
  }
 }
 state="getOperand2"
 decimal=false
}

function calculatePower(){
 if(state=="start"){
    ++ix
    state="getOperand2"
  }
 else{
   if(ix==0)
    ++ix
   else{
     var num=parseFloat(r[0])
     var exp=parseFloat(r[1])
     if(state=="changeSign")
      r[0]=Math.pow(num,-exp)
     else if(r[0].charAt(0)=="-" && r[1].charAt(0)=="0")
      r[0]="Complex Number"
     else
      r[0]=Math.pow(num,exp)
     displayResult(r[0])
    }
   }
  }

function findFactorial(x){
 var factorial=1
 if(x<0){
  alert('Check Input')
  x="ERROR"
 }
 else if(x==0 || x==1)
  r[ix]=factorial
 else
  {
   for(var i=1;i<=r[ix];i++)
    factorial*=i
    x=factorial
   }
 if(state=="start" || state==”gettingInteger”)
  x=factorial
 return x
 }

function applyFunction(){
 var selectionList=document.calculator.functions
 var selIX=selectionList.selectedIndex
 var sel=selectionList.options[selIX].value
 if(sel=="abs") r[ix]=Math.abs(r[ix])
 else if(sel=="acos") r[ix]=Math.acos(r[ix])
 else if(sel=="asin") r[ix]=Math.asin(r[ix])
 else if(sel=="atan") r[ix]=Math.atan(r[ix])
 else if(sel=="ceil") r[ix]=Math.ceil(r[ix])
 else if(sel=="cos") r[ix]=Math.cos(r[ix])
 else if(sel=="exp") r[ix]=Math.exp(r[ix])
 else if(sel=="floor") r[ix]=Math.floor(r[ix])
 else if(sel=="log") r[ix]=Math.log(r[ix])
 else if(sel=="sin") r[ix]=Math.sin(r[ix])
 else if(sel=="sqrt") r[ix]=Math.sqrt(r[ix])
 else if(sel=="fact") r[ix]=findFactorial(r[ix])
 else r[ix]=Math.tan(r[ix])
 decimal=false
 displayResult(r[ix])
}
// --></SCRIPT>
</HEAD>
<BODY>

<SCRIPT LANGUAGE="JavaScript"><!--
setStartState()
// --></SCRIPT>

<H1 ALIGN="CENTER">The Web Calculator</H1>
<FORM NAME="calculator">
<TABLE BORDER="BORDER" ALIGN="CENTER">
<TR>
<TD COLSPAN="4" ALIGN="CENTER"><INPUT TYPE="TEXT" NAME="total"
VALUE="0"
 SIZE="24"><BR><BR>
</TD></TR>
<TR>
<TD><INPUT TYPE="BUTTON" NAME="n0" VALUE="   0   "
 ONCLICK="addDigit(0)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n1" VALUE="   1   "
 ONCLICK="addDigit(1)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n2" VALUE="    2   "
 ONCLICK="addDigit(2)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="power"
 VALUE="  x^y " ONCLICK="calculatePower()"></TD>
</TR>
<TR>
<TD><INPUT TYPE="BUTTON" NAME="n3" VALUE="   3   "
 ONCLICK="addDigit(3)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n4" VALUE="   4   "
 ONCLICK="addDigit(4)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n5" VALUE="    5   "
 ONCLICK="addDigit(5)"></TD>
<TD COLSPAN="1" ROWSPAN="1"><INPUT TYPE="BUTTON"
 NAME="sign" VALUE="   +/-  " ONCLICK="changeSign()"></TD>
```

```
</TR>
<TR>
<TD><INPUT TYPE="BUTTON" NAME="n6" VALUE="   6   "
 ONCLICK="addDigit(6)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n7" VALUE="   7   "
 ONCLICK="addDigit(7)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="n8" VALUE="    8   "
 ONCLICK="addDigit(8)"></TD>
<TD COLSPAN="1" ROWSPAN="1"><INPUT
TYPE="BUTTON" NAME="pi" VALUE="   pi    "
 ONCLICK="setTo(Math.PI)"></TD>
</TR>
<TR>
<TD><INPUT TYPE="BUTTON" NAME="n9" VALUE="   9   "
 ONCLICK="addDigit(9)"></TD>
<TD><INPUT TYPE="BUTTON" NAME="decimal" VALUE="   .    "
 ONCLICK="addDecimalPoint()"></TD>
<TD><INPUT TYPE="BUTTON" NAME="plus" VALUE="   +    "
 ONCLICK="performOp('+')"></TD>
<TD COLSPAN="1" ROWSPAN="1"><INPUT TYPE="BUTTON" NAME="e"
 VALUE="    e    " ONCLICK="setTo(Math.E)"></TD>
</TR>
<TR>
<TD><INPUT TYPE="BUTTON" NAME="minus" VALUE="   -    "
 ONCLICK="performOp('-')"></TD>
<TD><INPUT TYPE="BUTTON" NAME="multiply" VALUE="    X  "
 ONCLICK="performOp('*')"></TD>
<TD><INPUT TYPE="BUTTON" NAME="divide" VALUE="    &divide;   "
 ONCLICK="performOp('/')"></TD>
<TD ROWSPAN="1"><INPUT
TYPE="BUTTON" NAME="clearField" VALUE="Clear"
 ONCLICK="clearDisplay()"></TD></TR>
<TR>
<TD COLSPAN="2" ROWSPAN="1" ALIGN="CENTER"><INPUT TYPE="BUTTON"
 NAME="random" VALUE="    Random    "
ONCLICK="setTo(Math.random())"></TD>
<TD COLSPAN="2" ROWSPAN="1" ALIGN="CENTER"><INPUT TYPE="BUTTON"
NAME="equals"
 VALUE="             =              " ONCLICK="calc()"></TD></TR>
<TR>
<TD COLSPAN="4" ROWSPAN="1" ALIGN="CENTER"><B>F(x):</B>
<SELECT NAME="functions" SIZE="1">
<OPTION VALUE="abs" SELECTED="SELECTED">abs(x)</OPTION>
<OPTION VALUE="acos">acos(x)</OPTION>
<OPTION VALUE="asin">asin(x)</OPTION>
<OPTION VALUE="atan">atan(x)</OPTION>
```

```
<OPTION VALUE="ceil">ceil(x)</OPTION>
<OPTION VALUE="cos">cos(x)</OPTION>
<OPTION VALUE="exp">exp(x)</OPTION>
<OPTION VALUE="fact">fact(x)</OPTION>
<OPTION VALUE="floor">floor(x)</OPTION>
<OPTION VALUE="log">log(x)</OPTION>
<OPTION VALUE="sin">sin(x)</OPTION>
<OPTION VALUE="sqrt">sqrt(x)</OPTION>
<OPTION VALUE="tan">tan(x)</OPTION>
</SELECT>
<INPUT TYPE="BUTTON" NAME="apply" VALUE="     Apply  F(x)   "
 onClick="applyFunction()"></TD></TR>
</TABLE>
</FORM>
<BR>
<DIV ALIGN="center">
<B>Click <A HREF="help.html">HELP</A> to know Input Operations</B>
</DIV>
<H6><I>Public Domain Software/2000</I>
<BR>Web Calculator : Beta Version 1.0  </H6>
</BODY>
</HTML>
```

APPENDIX 2
Slides

Slide 1: The home page of the Department Of Mathematical Sciences, Tezpur University as seen on a Microsoft Internet Explorer 5.0 browser



TEZPUR UNIVERSITY

The Department of Mathematical Sciences:

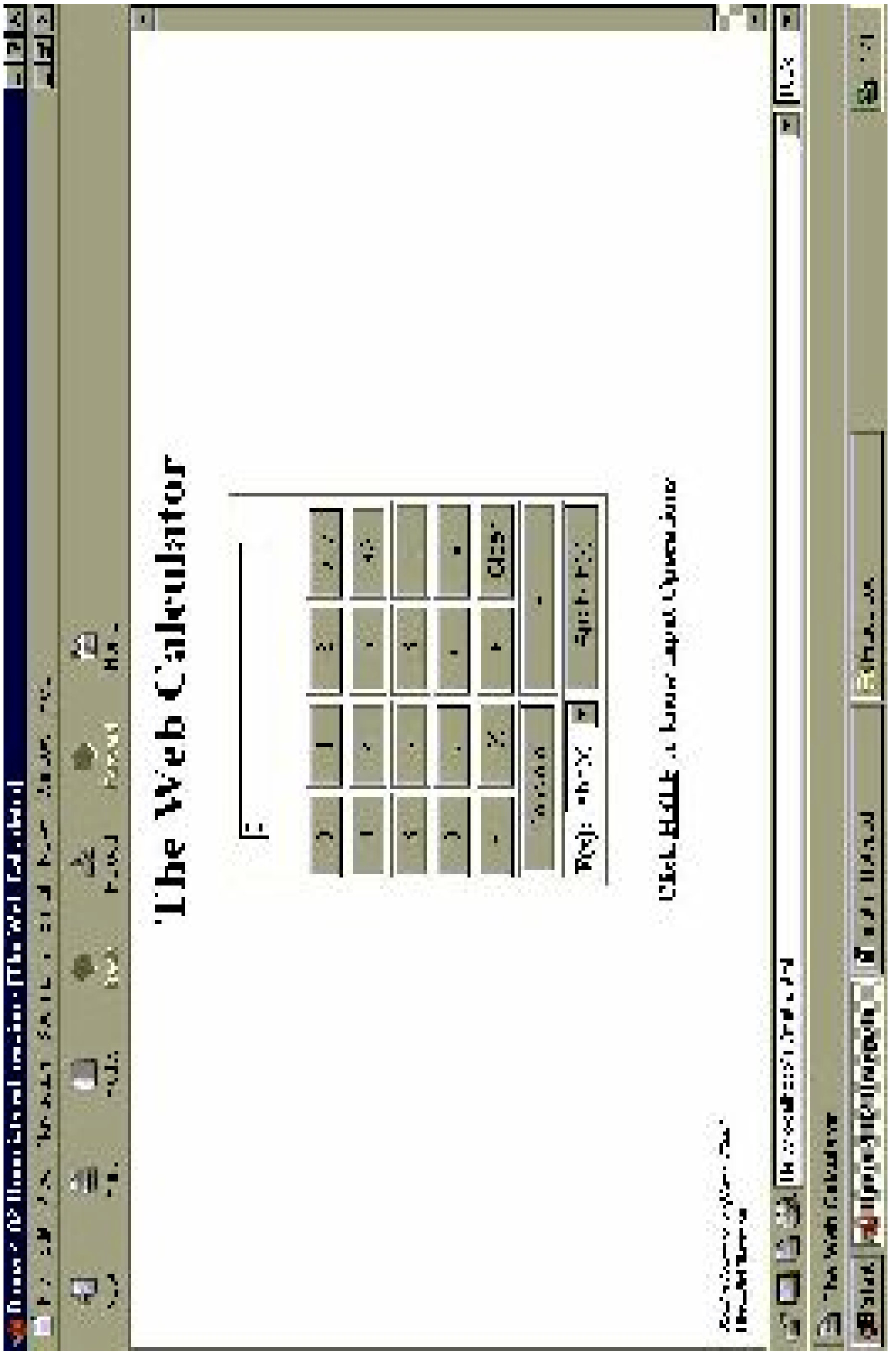
Tezpur University, the only [illegible] Central University of the region was established in [illegible] the Department of Mathematical Sciences on the 13th of [illegible] under the School of Science and Technology, Tezpur University.

The Department offers the courses
M.Sc/M.A. in Mathematics
Research Programme leading to Ph.D

These courses are offered in the following disciplines
Applied Mathematics
Pure Mathematics
Statistics
Theoretical Physics
Operations Research
Computational Mathematics

Date: [illegible]
Time: [illegible]

Slide 2: The Web Calculator as seen on an Opera 4.02 browser



APPENDIX 3
GLOSSARY

Applets: The Java programs that can be executed by the Web browsers.

ASCII (American Standard Code for Information Interchange): A character code used on the Internet. ASCII assigns each letters, digits, punctuation symbols, a unique sequence of binary digits. When textual data is transferred across the Internet, it is represented in ASCII.

Activex: It is an approach to developing and using software components that Microsoft created. ActiveX is an extension of the Object Linking and Embedding (OLE) and the Component Object Model (COM) technologies that were developed for Windows 3.1. OLE allows objects that were developed in one program to be displayed and edited in other. COM provides a general approach for software components to communicate and exchange services.

Browser: A program that displays documents on the Internet. Netscape Navigator/Communicator and Microsoft Internet Explorer are the most popular browsers.

CGI (Common Gateway Interface): The CGI is a standard that specifies how external programs may be used by Web servers. Programs that adhere to the CGI standard are referred to as CGI programs.

Client: A software application that works on our behalf to extract a service from a server somewhere on the network. For instance, a web browser is client that lets us view documents on remote servers.

Digital: Any technology that uses numbers to represent information. A computer is inherently digital because it represents keystrokes, pictures, texts and sounds using numbers.

FTP (File Transfer Protocol): The Internet service used to transfer a copy of a file from one computer to another. After contacting a remote computer, a user must enter a login name and password. Some FTP servers allow access to public files through the special login anonymous.

GIF (Graphic Interchange Format): Common file format of graphic images displayed on the Internet. It can store a maximum of 256 colors.

HTTP (HyperText Transfer Protocol): HTTP is the protocol used for communication between browsers and Web Servers. HTTP uses a request/response model of communication. A browser establishes a connection with a server and sends URL requests to the server. The server processes the browser’s request and sends a response back to the browser.

Hypertext: A method of presenting information so that the user can view it in a non-sequential way, regardless of how the topics were originally arranged. In a hypertext application, we can browse through the information flexibly, choosing to follow a new path each time we access the information.

JPEG (Joint Photographic Experts Group): A popular graphic file format on the Internet capable of storing up to 16 million colors.

Protocol: In networking and communications, the formal specifications that define the procedures to follow when transmitting and receiving data. Protocols define the format, timing, sequence, and error checking on the network.

Router: A device that forwards packets between networks, using information from the Network layer or from routing tables. The networks connected by a router may use the same or different networking principles.

Server: A networked computer that provides some kind of service or information. Internet servers are classified by the type(s) of information they offer, for e.g., mail server provides electronic mail. A single computer can act as a number of different servers.

TCP/IP (Transmission Control Protocol/Internet Protocol): Literally the name of protocols that specify how computers communicate on the Internet. Informally, the name refers to the software that implements the TCP/IP protocols. All computers that use the Internet need TCP/IP software. TCP handles the difficult task of ensuring that all data arrives at the destination in the correct order. IP on the other hand is a specification for the format of packets, computer use when they communicate across the Internet.

URL (Uniform Resource Locator): The notation used to specify the addresses of an Internet file or service.

W3C: The World Wide Web Consortium. A consortium of many companies and organizations that “exists to develop common standards for the evolution of the WWW”. It is run by a joint effort between the Laboratory for Computer Science at the Massachusetts Institute of Technology and CERN, the European Particle Physics Laboratory, where the WWW was first developed.

Appendix 4
References

From the Bookshelf:

- Coleman Pat, Dyson Peter, The ABCs of Intranets, First Indian Edition, 1997, BPB Publications, B-14, Connaught Place, New Delhi – 110001
- Corner Douglas E., The Internet Book, Second Indian Reprint, 1999, Prentice – Hall of India Private Limited, M – 97,Connaught Place, New Delhi – 110001
- Gillian Jason, Ting Charlton, Wyke R. Allen, Pure JavaScript, First Indian Edition, 1999, Techmedia, Munish Plaza, Darya Ganj, New Delhi – 110002
- Goodman Danny, Dynamic HTML - The Definitive Reference, First Indian Reprint, 1998, Shroff Publishers and Distributors Private Limited, M. K. Amin Marg, Fort, Mumbai – 400001
- Holzchlag Molly, Oliver Dick, Teach Yourself HTML 4 in 24 Hours, First Indian Edition,1998, Techmedia, Munish Plaza, Darya Ganj, New Delhi – 110002
- Jaworski James, Mastering JavaScript and JScript, First Indian Edition, 1999, BPB Publications, B-14, Connaught Place, New Delhi – 110001
- Mack E. Stephen, Platt Sanan, HTML 4.0 No Experience Required, First Indian Edition, 1998, BPB Publications, B-14, Connaught Place, New Delhi – 110001
- Merger Jason J., The World Wide Web, Mosaic and More, International Edition, 1995, Mc-Graw Hill Book Company, Singapore
- Summitt Paul M., Summitt Mary J., Creating Cool Interactive Web Sites, 1996, Comdex Computer Publishing, New Delhi - 110002

From the Web:

ü http://www.sybex.com
ü http://www.w3c.org
ü http://www.operasoftware.com
ü http://www.jasc.com
ü http://www.msn.com
ü http://www.home.netscape.com
ü http://www-sci.lib.uci.edu
ü http://www.tezu.ernet.in